Ph.D. DISSERTATION

Algorithms for Histogram Equalization in Image Enhancement and Link Prediction in Social Networks

화질 개선을 위한 히스토그램 평활화 및 소셜 네트워크에서의 링크 예측 알고리즘

BY

Hyoungjun Jeon

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Algorithms for Histogram Equalization in Image Enhancement and Link Prediction in Social Networks

화질 개선을 위한 히스토그램 평활화 및 소셜 네트워크에서의 링크 예측 알고리즘

BY

Hyoungjun Jeon

FEBRUARY 2017

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Algorithms for Histogram Equalization in Image Enhancement and Link Prediction in Social Networks

화질 개선을 위한 히스토그램 평활화 및 소셜 네트워크에서의 링크 예측 알고리즘

지도교수 김 태 환

이 논문을 공학박사 학위논문으로 제출함

2016년 11월

서울대학교 대학원

전기컴퓨터 공학부

전 형 준

전형준의 공학박사 학위 논문을 인준함

2016년 12월

위 원 장: _____

부위원장: _____

위 원: _____

위 원: _____

위 원: _____

Abstract

Data processing method is exploited to obtain the expected results by processing and analyzing the input data. In this research, we have focused on processing image and social network data in the area of image processing and social network analysis respectively by using data processing method. Gray-level context-driven histogram equalization and community-adaptive link prediction are proposed for image processing and social network analysis respectively. The abstractions of these two data processing methods are as follows.

First, histogram equalization, which redefines the distribution of gray-levels in an image, is an important step in image processing to enhance the image quality. Until now, numerous histogram equalization techniques have been proposed, among which the majority of them have focused on solving the problem of how the gray-levels in the histogram of an input image should be properly partitioned so that the image produced by collecting all equalization results for the partitioned sub-histograms leads to the quality enhancement of the image. However, the partition based equalization methods have an inherent limitation of not being able to equalize a sub-histogram crossing a partition boundary, which is the main cause of image distortion. In this work, we propose a gray-level context-driven histogram equalization method to overcome this limitation. In short, rather than constraining disjoint mapping ranges of the gray-levels among the partitions, we devise two enabling techniques: (1) a mapping range for each gray-level with no range-disjoint constraint and (2) a mapping distance between two adjacent gray-levels to make a full exploitation of mapping flexibility of gray-levels. We formulate the histogram equalization problem integrating the two techniques into a flow optimization problem in a specially designed structure of a network, and solve it globally and efficiently. In addition, we seamlessly combine the factor of power consumption into our network flow optimization formulation to make an easy trade-

off between image quality and power saving.

Second, Link prediction is one of hot research topics in social network analysis. Link prediction problem is to find a small set of node pairs in the networks that are not directly connected, but will be very likely to be connected in the future. To improve the prediction accuracy, many works have attempted to consider the community information, if available, in the social network structure. One common strategy of the prior community-aware link prediction algorithms is that they devised a sort of unified link prediction formulation that simply includes a premium term to express whether a link is structurally in the same community or not. However, since the formulation of the premium term relies on the structural formation of communities only, it cannot take into account the fact that the communities in different social networks, though they form almost identical community structures, can make different levels of influence on the link prediction. To cope with this limitation, we propose an adaptive approach, in which we use two separate link predictions depending on inter or intra-links in community, and then balance the links based on the degree of community influence on link prediction.

In conclusion, through experiments with the diverse datasets, it is shown that our proposed gray-level context-driven histogram equalization and community-adaptive link prediction are able to achieve much more improved performance compared to previous data processing methods in each of the image processing and social network analysis area.

keywords: Data processing, Image processing, Histogram equalization, Social network analysis, Link prediction

student number: 2013-30258

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Since the big data era unfolded, various types of data are increasingly emerging including unstructured data types, such as image, social network data, graph and so on. Recently, to exploit these various data type, the research of data processing method for a diverse area has become active to be in an important position [1, 2]. Data processing method is exploited to obtain the expected results by processing and analyzing the input data. It is exploited in various research areas and applications for processing and analyzing the input data.

In this research, we have focused on processing image and social network data in the area of image processing and social network analysis respectively by using data processing method. In the image processing area, we have proposed a method, called gray-level context-driven histogram equalization, based on histogram equalization which is one of famous image processing algorithms. A link prediction method, called a community-adaptive link prediction, has been proposed also as an important research area in the social network analysis.

1.2 Histogram Equalization in Image Enhancement

In image processing, histogram equalization (HE) [3] is a popular method to enhance the image quality and contrast by transforming input pixel intensities into output pixel intensities to make the histogram of the output image as uniform as possible. Due to its simplicity and effectiveness, HE is widely used in various applications, including digital photography, radar, and medical images. Formally, if we let $H(k)$ denote the number of pixels of gray-level k in $\{0, 1, \cdots, L-1\}$ in an input image, histogram equalization maps the gray-level k of the $H(k)$ pixels into a gray-level $x(k)$. Thus, the resultant image quality is determined by the mapping function $x(\cdot)$ used in the histogram equalization.

As shown in Fig. 1.1, histogram equalization could improve the contrast of image by redefining the distribution of gray-levels. However, despite several merits of histogram equalization such as simple processing, histogram equalization has problems that should be overcomed. This straight use of histogram equalization may change the original brightness of an input image, and produce annoying artifacts, or deteriorate visual quality as shown in Fig. 1.2. To overcome these limitations, many researchers have proposed methods based on histogram equalization.

1.3 Link Prediction in Social Networks

In recent big data era, social network analysis is becoming an important research area with various types of emerging social networks. Especially, link prediction method, which is one of the main issue on the social network analysis, has been increasingly researched [4, 5]. The link prediction is the method on predicting vertex pairs that a link emerges between them at future time t' but not existing edge at current time t by analyzing the network at time t. There are a variety of applications for link prediction such as recommendation system of partners or friends, analysis system of the

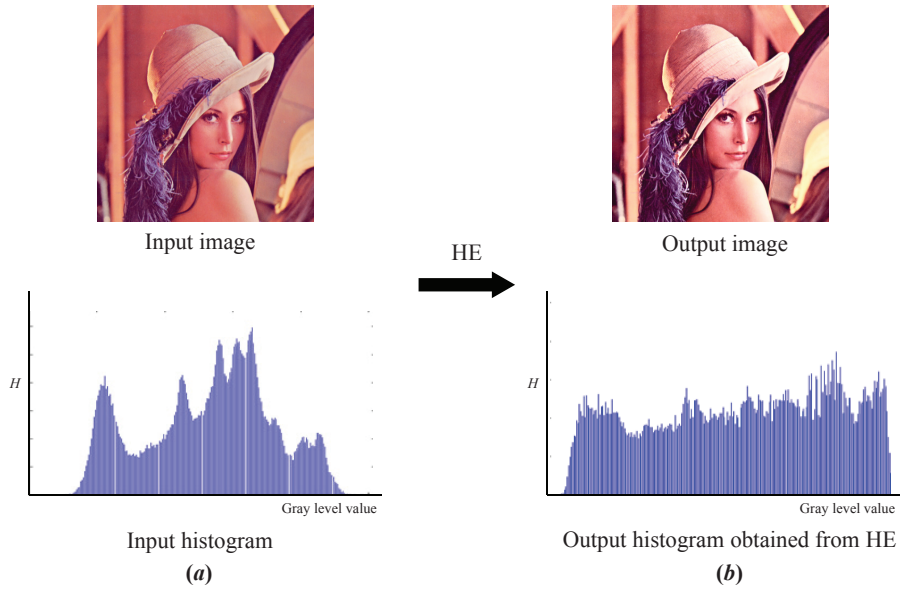


Figure 1.1: Example showing the contrast enhancement of input image processed by a histogram equalization.

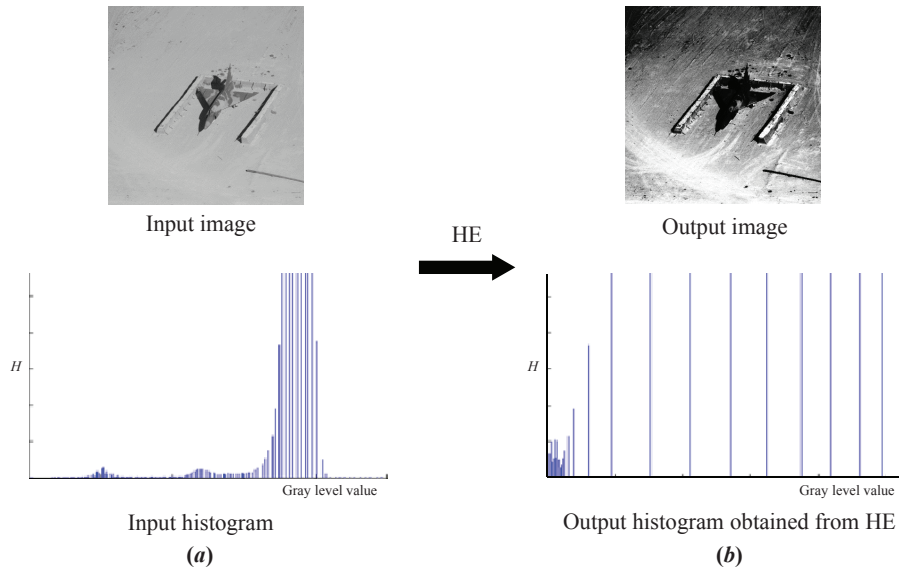


Figure 1.2: Example showing the image distortion of a histogram equalization.

correlation between humans and exploration of the experts or collaborations between researchers in the academia [6]. These applications have been expanded increasingly as the area of social networks grows.

Similarity-based link prediction is the most popular method for solving the problem [7, 8]. Given the network, node pairs which don't have edges between them are scored as their similarity to predict the link at future time. Each vertex pair of v_i and v_j is assigned $score(i, j)$ which means the possibility of appearing link between them. Links could be predicted by choosing vertex pairs with high score. Thus, after the completion of link prediction process, each node pair is sorted by decreasing order so as to select the node pairs of high $score(i, j)$ as shown in Fig. 1.3.

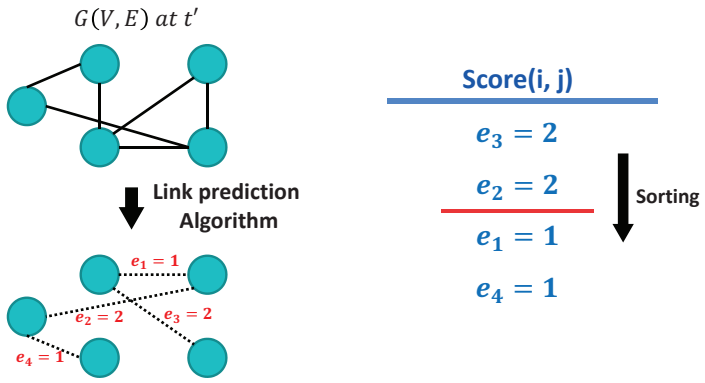


Figure 1.3: Example showing a process of similarity-based link prediction algorithm.

Chapter 2

Gray-Level Context-Driven Histogram Equlaization

2.1 Introduction

The basic idea of using histogram equalization is to stretch the range of gray-levels in input image to enhance the image quality and contrast. However, this straight use of histogram equalization may change the original brightness of an input image, and produce annoying artifacts, or deteriorate visual quality. This is because the mean brightness of the transformed output image approaches to the middle gray-level regardless of the mean brightness of the input image. To preserve the mean brightness while enhancing visual quality, many researchers have employed a divide-and-conquer approach, which partitions the range of the whole gray-levels into several sub-ranges and applies equalization to the histograms of the sub-ranges separately. The main difference among the existing divide-and-conquer based histogram equalizations is how they partition the input histogram.

Kim [9] proposed a histogram equalization, called *brightness preserving bi-histogram equalization* (BBHE), which divides an input histogram into exactly two parts where the cutting point is the gray-level corresponding to the mean brightness value of the input image. Later, Wang, Chen, and Zhang [10] formulated the measurement of the

image information that can be obtained by human eyes into the entropy equation with gray-level distribution probability of the image as an input parameter. Their proposed histogram equalization, called *dualistic sub-image histogram equalization* (DSIHE) showed that the entropy is maximized when the input histogram is partitioned into two parts where the cutting point is the gray-level in which the value of its cumulative distribution probability is 1/2. Chen and Ramli [11] improved BBHE [9] by selecting the cutting gray-level that minimizes the difference of the mean brightness values of the two partitioned sub-images. Rather than partitioning into two parts, Wadud *et al.* [12] divided an input histogram into more than two parts. Their algorithm, called *dynamic histogram equalization* (DHE), selected as cutting points the gray-levels of local minima on the histogram. Then, Ibrahim and Kongi [13] enhanced DHE [12] to take into account the brightness preservation of the transformed image. Sheet *et al.* [14] further elaborated DHE [12] by using a fuzzy statistical analysis of an image to improve the accuracy of the selection of cutting gray-levels. Celik and Tjahjadi [15] used the Gaussian mixture model for partitioning the dynamic range of the input image. Singh and Kapoor proposed several sub-image based histogram equalizations: *exposure based sub-image histogram equalization* (ESIHE) [16] which divides the original image into sub-images based on intensity levels, *median-mean based sub-image-clipped histogram equalization* (MMSICHE) [17] which divides clipped histogram to several sub-images by mean intensity, and recursive version of ESIHE [16] histogram equalization [18] which enhances low exposure images.

Besides histogram equalization, there are several contrast enhancement methods based on histogram modification framework (e.g., [19, 20, 21]). Gu *et al.* implemented so-called *general histogram modification framework* (GHMF) [19] by using hybrid transformation technique. They also proposed an automatic contrast enhancement technique with a complete histogram modification framework [20]. In addition, they proposed a metric called *reduced-reference image quality metric for contrast*

change (RIQMC), based on which they proposed an optimal histogram mapping (ROHIM) [21].

The fundamental limitation of the histogram partition based equalization methods is that the equalization cannot be applied to a sub-histogram across a cutting gray-level, which is the main cause of image distortion. In this work, we overcome this limitation by devising two key concepts: (1) *mapping range on the gray-level basis* and (2) *mapping distance between gray levels*, which completely resolve the inability problem of the partition based equalization methods for equalizing the sub-histograms across cutting gray-levels. Note that various non-partition based histogram equalization methods (e.g., [22, 23, 24, 25, 27]) have been proposed to overcome the limitation of the partition based ones. Celik and Tjahjadi [22] equalized the histogram by using the sum of Frobenius norms. Lai *et al.* [23] focused texture region histogram equalization to suppress the impact of smoother regions. Poddar *et al.* [24] adaptively modified input histogram to be spike free to improve the contrast of the image. On the other hand, Lee, Lee, and Kim [25] transformed input histogram into a log-based histogram (LHM) [26] to make the portions of sharp increase/decrease curve on the gray intensity distribution smooth. Ghita, Ilea, and Paul [27] proposed a method called *texture enhanced histogram equalization* (TE-HE) which performed, after decomposition, in a way to minimize total variation with a L^a norm (TV-L^a, where $a = 1$).

However, the non-partition based methods have difficulty in consistently maintaining the original image brightness on a diverse spectrum of images, even though the image contrast is enhanced. However, our proposed gray-level based mapping range technique is able to control the image brightness very conveniently and effectively.

Fig. 2.1 shows the algorithms flow of our gray-level context-driven histogram equalization algorithm. The rest of the chapter is organized in order of our algorithms flow as follows. In section 2.2, we propose two new histogram enabling techniques which will be used effectively to overcome the limitation of the existing partition based

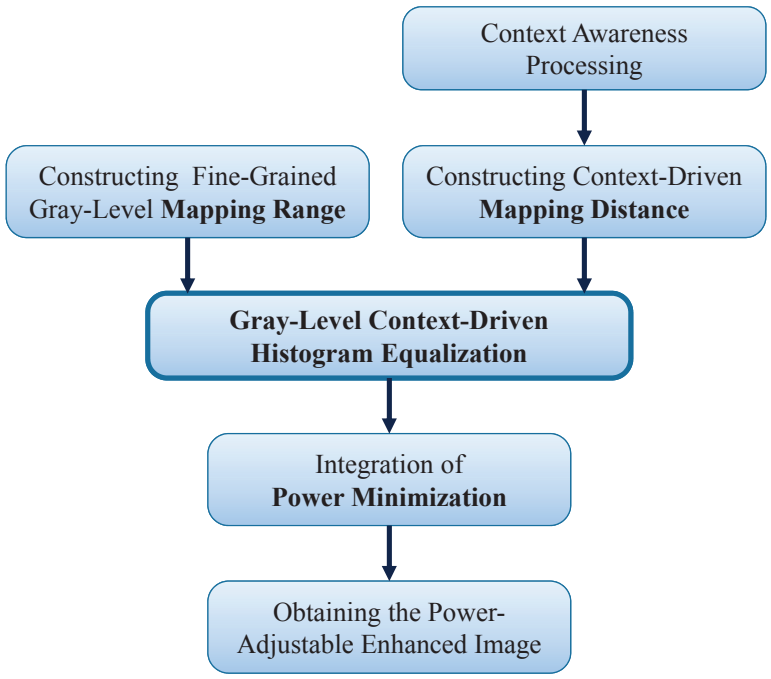


Figure 2.1: Alogrithm flow of our gray-level context-driven histogram equalization algorithm.

histogram equalization methods. Then, section 2.3 describes our histogram equalization algorithm integrating the two techniques, followed by proposing an extension of the algorithm to support the factor of minimizing power consumption in section 2.4. Experimental results to demonstrate the effectiveness of the proposed equalization method is provided in section 2.5. Finally, a conclusion of the work is given in section 2.6.

2.2 Enabling Techniques for Equalization

The next subsections describe two new equalization enabling techniques called *gray-level mapping range* and *context-driven mapping distance*, based on which our histogram equalization algorithm is built.

2.2.1 Fine-Grained Gray-Level Mapping Range

Let $\{0, 1, \cdots, L-1\}$ be the set of gray-levels and $H(i)$ be the intensity (i.e., pixel count) of a gray-level $i \in \{0, \cdots, L-1\}$ in image. We use $x(\cdot)$ to denote the mapping function of a histogram equalization. Then, any mapping function of equalization should satisfy the mapping order constraint:

Definition 1 (**Mapping order constraint**): *if $i < j$, then $x(i) \leq x(j)$, for every i and j in $\{0, \cdots, L-1\}$.*

We formally define mapping range as:

Definition 2 (**Mapping range**): *Mapping range, $R(i)$, of an equalization algorithm for gray-level i is the gray-level interval to which the equalization can feasibly map i.*

The $R(i)$ interval varies depending on the equalization algorithms employed. For example, if the bi-histogram equalization algorithm (BBHE) is used, $R(i) = [0, g_{mean}]$ for $i = 0, \cdots, g_{mean}$ and $R(i) = [g_{mean} + 1, L-1]$ for $i = g_{mean} + 1, \cdots, L-1$ where g_{mean} represents the gray-level corresponding to the mean brightness of the

input image.

The histogram partition based equalization algorithms are inherently lack of controlling two kinds of image quality: (1) image smoothness and (2) image distortion.

- *Image smoothness*: Let $k \in \{0, 1, \cdots, L-1\}$ be a gray-level which cuts the histogram of an input image according to the applied equalization algorithm. Then, the mapping range $R(\cdot)$ of the gray-levels as close as to k would be very tight because they are disallowed to be mapped to gray-levels across k, which results in failing to smooth the image by the equalization. For example, Fig. 2.2(a) shows the input image and its histogram while Fig. 2.2(b) shows the histogram and the corresponding image produced by the application of BBHE to the input histogram where g_{mean} acts as k. The image region failing smoothness is indicated by the red circle and the corresponding gray-levels are also marked in Fig. 2.2(b).

- *Image distortion*: If the majority of pixel counts of gray-levels is uniformly distributed to the partitioned histograms, the distance between the mapped gray-levels of the majority pixels can be unnecessarily long or short. This is because mapping ranges $R(\cdot)$ of gray-levels in different partitions are disjoint. For example, Fig. 2.3(a) shows the input image and its histogram while Fig. 2.3(b) shows the histogram and the corresponding image produced by the application of BBHE to the input histogram. As specified in the histogram in Fig. 2.3, since for a pair of gray-levels g_{low} and g_{high} such that $g_{low} < g_{mean} < g_{high}$, g_{low} cannot be mapped to a gray-level in $R(g_{high})$ i.e., $x(g_{low}) \notin R(g_{high})$ and g_{high} cannot be mapped to a gray-level in $R(g_{low})$ i.e., $x(g_{high}) \notin R(g_{low})$, the image is considerably distorted by BBHE, as shown in Fig. 2.3(b).

The main cause of the inability of acquiring consistent image smoothness and image naturalness in the conventional histogram partition based equalizations is that ev-

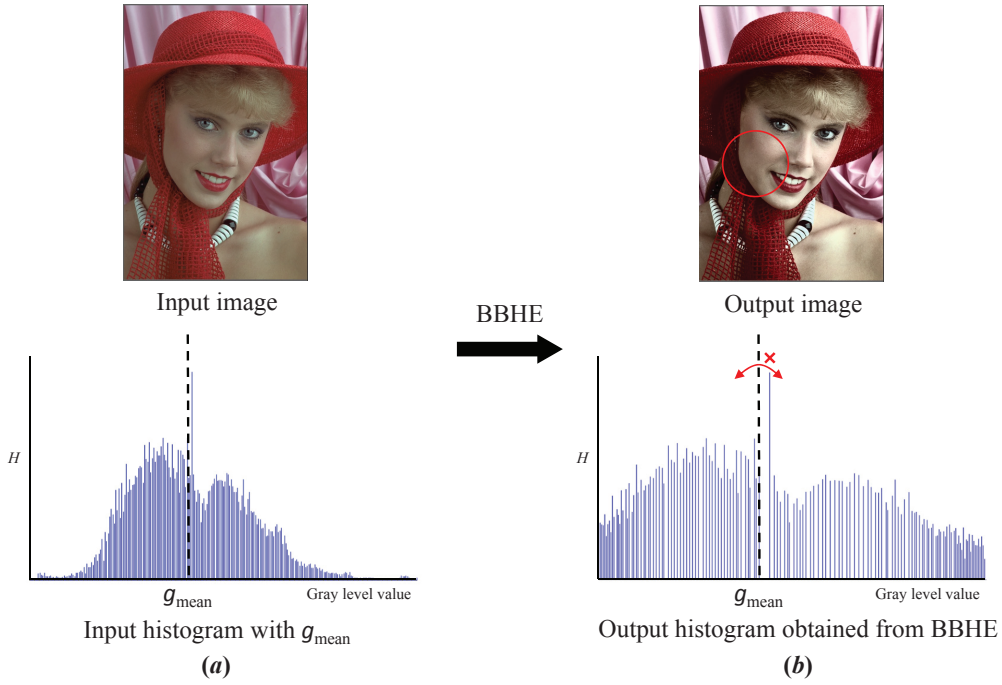


Figure 2.2: Example showing the image non-smoothness of a partition based histogram equalization.

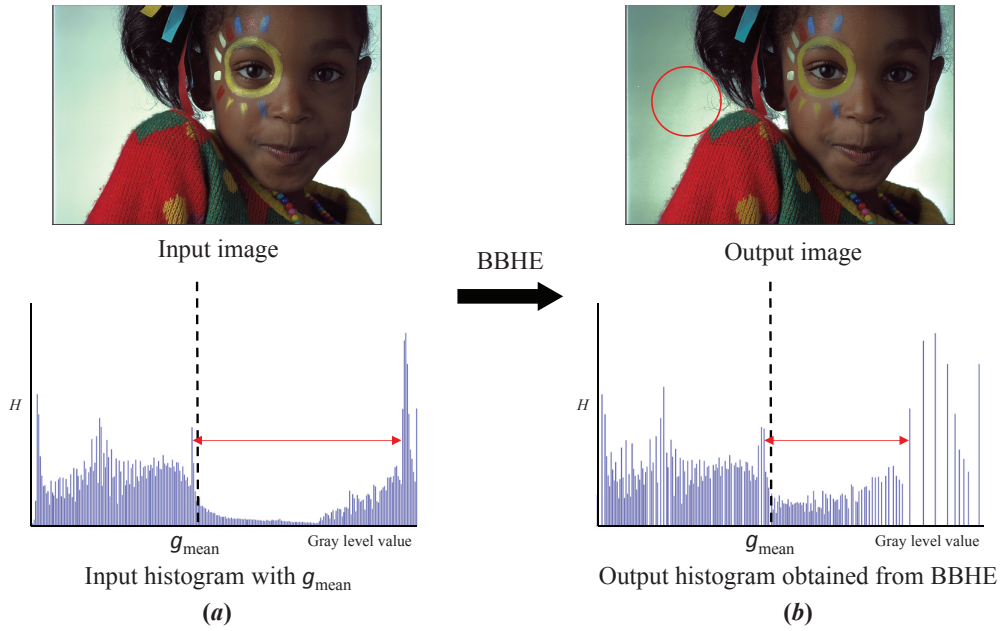


Figure 2.3: Example showing the image distortion of a partition based histogram equalization.

ery gray-level in a partitioned histogram is constrained not to be mapped, by equalization, to a gray-level in another partitioned histogram. To overcome the inability, rather than generating the mapping range $R(\cdot)$ of every gray-level exclusively based on the partitioned histogram it belongs to, we want to compute the mapping range $R(\cdot)$ based on the relative degree of its intensity on the image. Formally, the mapping range $R(i)$ of a gray-level i is of the form:

$$R(i) = [i - \Delta_i, \;\; i + \Delta_i] \tag{2.1}$$

and Δ_i is computed by

$$\Delta_i = R_{UB} \times e^{-\alpha \times \frac{H(i)}{N}} \tag{2.2}$$

where R_{UB} is the upper bound of the length of $R(\cdot)$, $H(i)$ is the pixel count i.e., an intensity of i, and N is the total number of pixels in the image. α is a control parameter. Thus, the parameter R_{UB} could control the degree of contrast enhancement. As the value of R_{UB} increases, the processed image becomes close to the image from HE, whereas as the value of R_{UB} decreases, the image is close to the original image.

Fig. 2.4(a) illustrates the mapping range examples proposed by our formulation. The inverse logarithmic expression in $Eq.\ 2.2$ constraints that as the intensity of a gray-level decreases, the interval of its mapping range exponentially increases, as shown in Fig. 2.4(b).

2.2.2 Context-Driven Mapping Distance

The translation of an input image into a histogram is a kind of onto function, and not one-to-one function. That means, for a particular histogram, there exists always an image that can be translated into the histogram, but if two histograms translated from two images are exactly the same, the two images are not necessarily identical. Thus, two different input images may be translated into the same histogram, which is a fundamental limitation of the conventional histogram equalization methods since

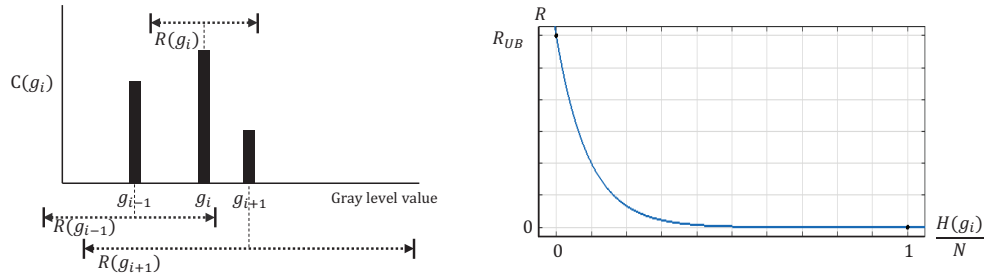


Figure 2.4: An illustration of the proposed mapping range of gray-levels. (a) Mapping range examples. (b) Mapping range curve with respect to the change of the intensity of gray-levels.

they perform equalization entirely based on the histogram of the input image and never know or exploit the image details (i.e., image context).

A few works (e.g., [28, 29]) tried to overcome the drawback of the context-unawareness in histogram equalization. Their common strategy is to partition the input image into several sub-images and apply equalization to each sub-image independently. The difference among the works is how they generate sub-images. However, this strategy still suffers image distortion since it is not able to consider balancing the equalization among sub-images.

Contrary to the image partitioning based histogram equalization approach, we want to implant the information of image context to the intensity bars in the histogram. For example, Fig. 2.5(a) shows the two intensity bars of gray-levels i and j of a histogram H (We assume that the intensity of every gray-level in between i and j is 0.) Then, the conventional equalization will stretch the intensity bars by considering the intensity values (i.e., bars' heights), regardless of the constituents of the intensities. Figs. 2.5(b) and 2.5(c) respectively show the constituents of the intensities of two different input images, say I_0 and I_0', which are translated into the histogram H $(= H(I_0) = H(I_0'))$ in Fig. 2.5(a). Fig. 2.5(b) shows that both of the intensity bars of i and j are contributed

by image objects A and B in I_0. This implies that the equalization should not take the two bars too apart because otherwise, the visual quality of objects A and B will be deteriorated. On the other hand, Fig. 2.5(c) shows that the intensity bars of i and j are contributed by quite different image objects. Thus, the equalization can stretch the histogram in a way to take the two bars too apart to enhance the image quality and contrast.

To implement the idea introduced in Fig. 2.5, it requires to solve two issues: (1) *How can we associate the intensity bars in histogram with image objects (i.e., image context)?* and (2) *How can we exploit the object information attached to the intensity bars in equalization?*

1. *Associating histogram with image context*: We scan the pixels in the image line-by-line, from the top to the bottom. In a line, we scan pixels from the left to the right. During the scanning process, we identify objects by comparing the gray-level of the current scan pixel with the gray-levels of the surrounding pixels. Let us suppose that, for simplicity, O_k is an object found (i.e., formed) so far just before when we are to scan pixel p_i and assume that p_j is one of the surrounding pixels of p_i and object O_k contains p_j. Note that every object, say O_i, extracted during the scanning process is represented with the pixel positions in O_i and the maximum and minimum gray-levels, denoted by $g_{max}(O_i)$ and $g_{min}(O_i)$, of the pixels in O_i. Then, the decision on whether p_i should be added to object O_k or not is guided by checking the following properties and constraints: (Let O'_k be the updated object of O_k by including p_i to O_1.)

   – *Object's global property* (Λ_g): O'_k should satisfy

   $$|g_{max}(O'_k) - g_{min}(O'_k)| \leq \Lambda_g. \tag{2.3}$$

   – *Object's local property* (Λ_l): O'_k should satisfy

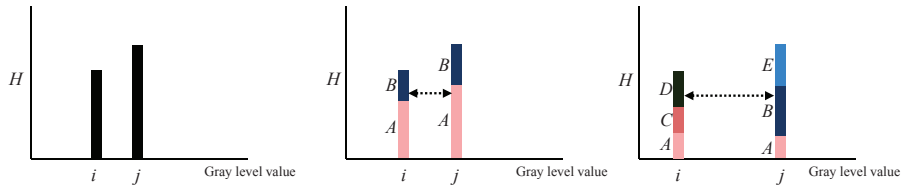


Figure 2.5: An illustration of the intensity representations. (a) Histogram H to be used in the conventional context unaware equalization. (b) The context-aware histogram for input image I_0 where the intensity bars of gray-levels i and j both are contributed by objects A and B in the image. A short distance between the two bars should be made by equalization to avoid the image quality distortion by the objects. (c) The context-aware histogram for input image I_0' where the intensity bars of gray-levels i and j are contributed by different image objects. A loose distance between the two bars can be made by equalization to enhance visual quality.

$$|g(p_i) - g(p_j)| \leq \Lambda_l \tag{2.4}$$

where $g(p_i)$ and $g(p_j)$ are the gray-levels of pixels p_i and p_j, respectively.

– *Object's local distance constraint* (Λ_d): O'_k should satisfy

$$|loc(i) - loc(j)| \leq \Lambda_d \tag{2.5}$$

where $loc(p_i)$ and $loc(p_j)$ are the positions of p_i and p_j, respectively.

– *Object's area constraint* (Σ):

$$|O'_k| \geq \Sigma \tag{2.6}$$

where $|O'_k|$ indicates the number of pixels in object O'_k. Note that initially each pixel is counted as a distinct object of size 1. After all scannings are done, we will remove the objects which violate the area constraint in $Eq.$ 2.6.

Here, Λ_g, Λ_l, Λ_d, and Σ are controlling parameters. The parameters could control the number and size of object clusters. The property of the parameters is:

Λ_g : As the value of Λ_g increases, the cluster size and computation overhead increase, and the degree of contrast enhancement decreases. On the other hand as the value of Λ_g decreases, the cluster size, computation overhead and image quality decrease.

Λ_l : As the value of Λ_l increases, the cluster size increases and the degree of contrast enhancement decreases, while as the value of Λ_l decreases, the cluster size and image quality decrease.

Λ_d : It prevents a large gradient region from binding to one cluster.

Σ : As the value of Σ increases, the number of clusters, computation overhead and image quality decrease. On the other hand as the value of Σ decreases, the number of clusters and computation overhead increase, and the degree of contrast enhancement decreases.

For each, p_j, of the surrounding pixels of the currently scanned pixel p_i, we check if the global and local properties and distance constraint are satisfied for the p_j's object updated by the inclusion of p_i. If more than one object satisfy all the properties and constraint, we choose the biggest object for the inclusion of the current pixel.

Fig. 2.6(a) shows an example of scanning pixels to extract objects, in which the numbers indicate the gray-levels of the pixel and the objects correspond to the groups of the numbers with the same color. (Each number with black color is the object with a single pixel.) Fig. 2.6(b) shows an input image and the extracted objects from the image, marked with different colors, and the resulting histogram embedding object information.

2. *Equalizing constraint by image context*: We define a correlation measurement, $\rho(i,j)$, between two adjacent intensity bars $H(i)$ and $H(j)$ in the histogram:

$$COUNT(i\backslash j) = \sum_{O_l \ s.t. \ p_t \in O_l, g(p_t)=j} |\{p_k | g(p_k) = i, p_k \in O_l\}| \tag{2.7}$$

$$COUNT(j\backslash i) = \sum_{O_l \ s.t. \ p_t \in O_l, g(p_t)=i} |\{p_k | g(p_k) = j, p_k \in O_l\}| \tag{2.8}$$

$$\rho(i,j) = \sqrt{\frac{COUNT(i\backslash j)}{H(i)} \times \frac{COUNT(j\backslash i)}{H(j)}} \tag{2.9}$$

where $COUNT(i\backslash j)$ represents the total number of pixels of gray-level i in the objects which also contain pixels of gray-level j. $COUNT(j\backslash i)$ is similarly

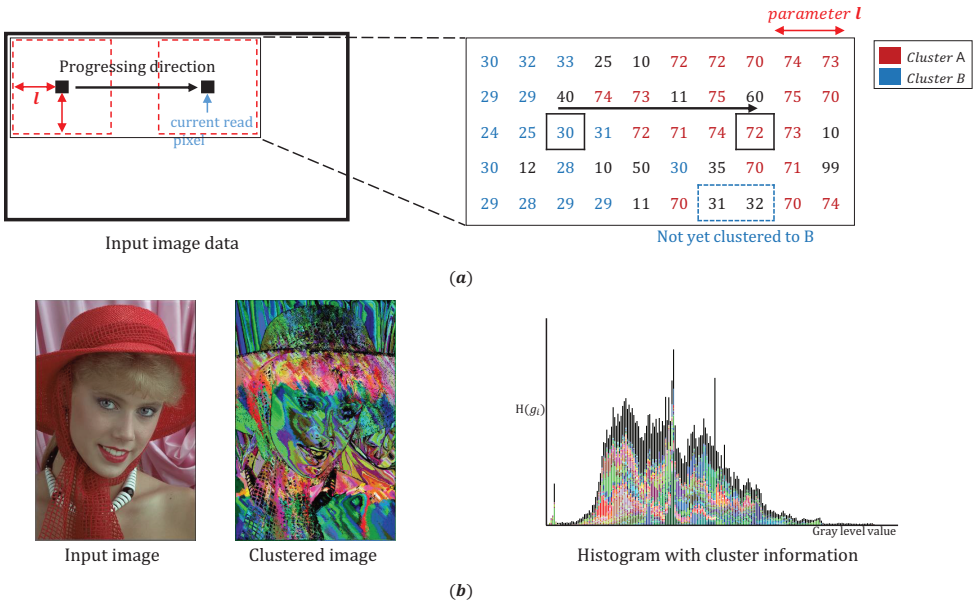


Figure 2.6: An illustration of cluster (object) extraction process and derivation of histogram embedding object information. (a) Object extraction process. (b) Object extraction and resulting histogram.

defined. Thus, $\rho(i,j)$ can be used to measure the degree of sharing gray-levels i and j in objects. We can easily check $0 \leq \rho(i,j) \leq 1$. If the value of $\rho(i,j)$ is close to 1, it is very likely that the objects either contain pixels of both gray-levels i and j or do not contain both. Thus, a tight distance constraint between the two intensity bars of i and j should be imposed on the equalization. On the other hand, if the value of $\rho(i,j)$ is close to 0, a loose distance constraint can be set. A formal definition of distance constraint can be described as:

Definition 3 (**Mapping distance**): *Minimum and maximum mapping distances, $D^{min}(i,j)$ and $D^{max}(i,j)$, of an equalization algorithm for gray levels i and j with $H(k) = 0$, $k = i+1, \cdots, j-1$ are the lower and upper bounds of $|x(i) - x(j)|$ where $x(i)$ and $x(j)$ represent the gray-levels to which the equalization can feasibly map i and j, respectively.*

The conventional histogram partition based equalization algorithms uniformly set $D^{min}(i,j) = |j-i|$ and $D^{max}(i,j) = g_{max} - g_{min}$ where g_{max} and g_{min} are the maximum and minimum gray-levels of the partition in which i and j are contained. Therefore, no image context is taken into account in the formulation of mapping distances. Instead, we make use of the correlation measurement $\rho(i,j)$ in the formulation of mapping distance $D(i,j) = [D^{min}(i,j), D^{max}(i,j)]$:

$$D^{max}(i,j) = |j-i| + (-D_{UB} \times \rho(i,j) + D_{UB}), \tag{2.10}$$

$$D^{min}(i,j) = |j-i| \times \rho(i,j), \tag{2.11}$$

which satisfy

$$D^{min}(i,j) \leq |x(i) - x(j)| \leq D^{max}(i,j) \tag{2.12}$$

where D_{UB} is a parameter to be used to set the largest one among $D^{max}(\cdot,\cdot)$ values in the histogram. Thus, the D_{UB} could control the distance between its adjacency histogram bars. As the value of D_{UB} increases, the noise increases and naturalness of the original image decreases.

Note that if two adjacent non-zero intensity bars, say $H(i)$ and $H(j)$, in input histogram have many number of pixels in common objects, it will increase the value of $\rho(i,j)$, which in turn decreases the value of $D^{max}(i,j)$ and increases the value of $D^{min}(i,j)$. As a result, the equalization will constrain the two intensity bars not to be placed far apart.

Fig. 2.7(a) illustrates the mapping distance examples proposed by our formulation. As a correlation between two adjacency histograms gets stronger, the interval of their maximum mapping distance D^{max} in $Eq.$ 2.10 linearly decreases as shown in Fig. 2.7(b).

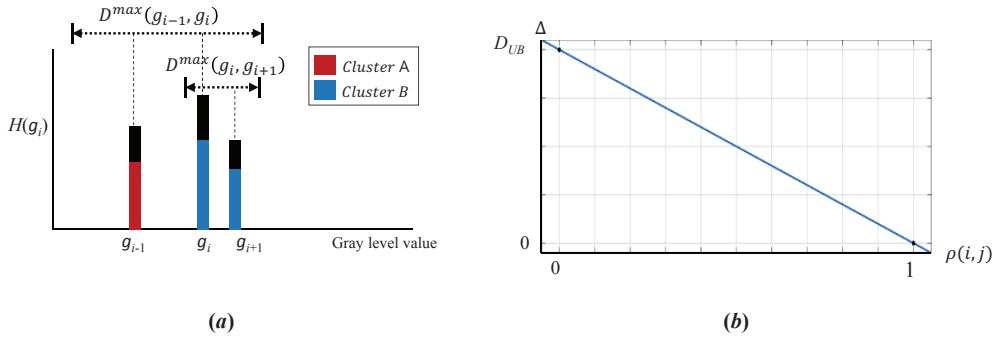


Figure 2.7: An illustration of the proposed mapping range of gray-levels. (a) Mapping distance examples. (b) Mapping distance curve with respect to the change of the correlation between adjacency histogram bars.

2.3 The Network Flow Formulation

The histogram equalization problem we want to solve can be described as:

Problem 1: *For an input image I_0 with gray-levels in $\{0, 1, \cdots, L-1\}$, its histogram H_0 and a reference equalization (mapping) function $x_0(\cdot)$, our proposed gray-level context-driven Histogram Equalization algorithm called* HE-gc *performs the two steps: (Step 1) computing mapping ranges and mapping distances from I_0 and H_0 and (Step 2) finding an equalization function $x(\cdot)$ which minimizes the quantity of*

$$\Delta_{diff} = \sum_{i=0}^{L-1} |x(i) - x_0(i)| \tag{2.13}$$

while satisfying the mapping order, mapping range, and mapping distance constraints.

Step 1 is to compute the values of mapping range $R(\cdot)$ of gray-levels according to $Eq.$ 2.1 and the values of mapping distances $D^{max}(\cdot,\cdot)$ and $D^{min}(\cdot,\cdot)$ according to the definitions in $Eq.$ 2.10 and $Eq.$ 2.11. Step 2 is to find a mapping function of gray-levels under the constraints of mapping distance and mapping range as well as mapping order. We formulate the problem of Step 2 into a flow optimization problem in a network $G(V, A, W)$ and solve it optimally in polynomial time. The construction of $G(V, A, W)$ is as follows:

1. For each gray-level $i \in \{0, \cdots, L-1\}$, we create $|R(i)|$ number of distinct nodes and arrange them vertically in a column. Thus, the network has at most L columns of nodes as shown in Fig. 2.8 where the numbers on the top of dotted box indicate the corresponding gray-levels of the column. (For gray-levels with intensity of 0, there is no need to create column.) For example, in Fig. 2.8 for mapping range $R(0) = [0, 4]$ of gray-level 0, we create nodes $n_{0,0}$, $n_{0,1}$, $\cdots$, $n_{0,4} \in V$ in the leftmost column, which correspond to feasible mappings of gray-level 0 to gray-levels 0, 1, $\cdots$, and 4 by equalization function $x(\cdot)$, respectively, and for mapping range $R(65) = [56, 74]$ of gray-level 65, we create 19

(= 74-56+1) nodes $n_{65,56}$, $n_{65,57}$, $\cdots$, $n_{65,74} \in V$ in the middle column, which correspond to feasible mappings of gray-level 65 to 56, 57, and 74 by equalization function $x(\cdot)$, respectively. Besides, we include two dummy nodes *Source* and *Sink* in V.

2. We include an arc from *Source* to every node in the first column of G to arc set A, and include an arc from every node in the last column to *Sink* to A. For every pair of nodes n_{i,k_1} and n_{j,k_2} between two adjacent columns of gray-levels i and j we add an arc $n_{i,k_1} \rightarrow n_{j,k_2}$ to A if $D^{min}(i,j) \leq k_2 - k_1 \leq D^{max}(i,j)$. For example, in Fig. 2.8 mapping distance $[D^{min}, D^{max}]$ of every pair of two adjacent gray-levels i and j is shown at the bottom. It shows that arc $(n_{0,0} \rightarrow n_{1,1})$ is in $\in A$ since $D^{min}(0,1) = 0 \leq 1(= 1-0) \leq D^{max}(0,1) = 3$, but there is no arc of $(n_{0,1} \rightarrow n_{1,6})$ since $D^{min}(0,1) = 0 \leq 5(= 6-1) \not\leq D^{max}(0,1) = 3$.

3. We assign a weight $w_{i,k} \in W$ to each node $n_{i,k}$. $w_{i,k}$ is set to $|k - x_0(i)|$ where $x_0(i)$ is a gray-level to which gray-level i is transformed by the reference histogram equalization. We assign *Source* and *Sink* with weight 0.

Then, we find a path from *Source* to *Sink* whose total sum of the weights of the nodes on the path is minimal. For example, the minimal cost path is in Fig. 2.8 is shown in red color, which is *Source* $\rightarrow n_{0,3} \rightarrow n_{1,6} \rightarrow \cdot \rightarrow n_{65,73} \cdots \rightarrow n_{244,241} \cdots \rightarrow n_{255,254} \rightarrow$ *Sink*. The path defines mapping function $x(\cdot)$ such that $x(0) = 3$, $x(1) = 6$, $\cdots$, $x(65) = 73$, $\cdots$, $x(244) = 241$, $\cdots$, $x(255) = 254$. Since the creation of nodes in G implements the mapping range constraint, the creation of arcs implements the mapping distance and mapping order, and finding a minimal cost path in G corresponds to the minimization of Δ_{diff} cost in $Eq.$ 2.13, we can solve the problem by using a shortest path algorithm in $G(V, A, W)$.

We used the Bellman-Ford algorithm whose time complexity is bounded by $O(|V||A|)$

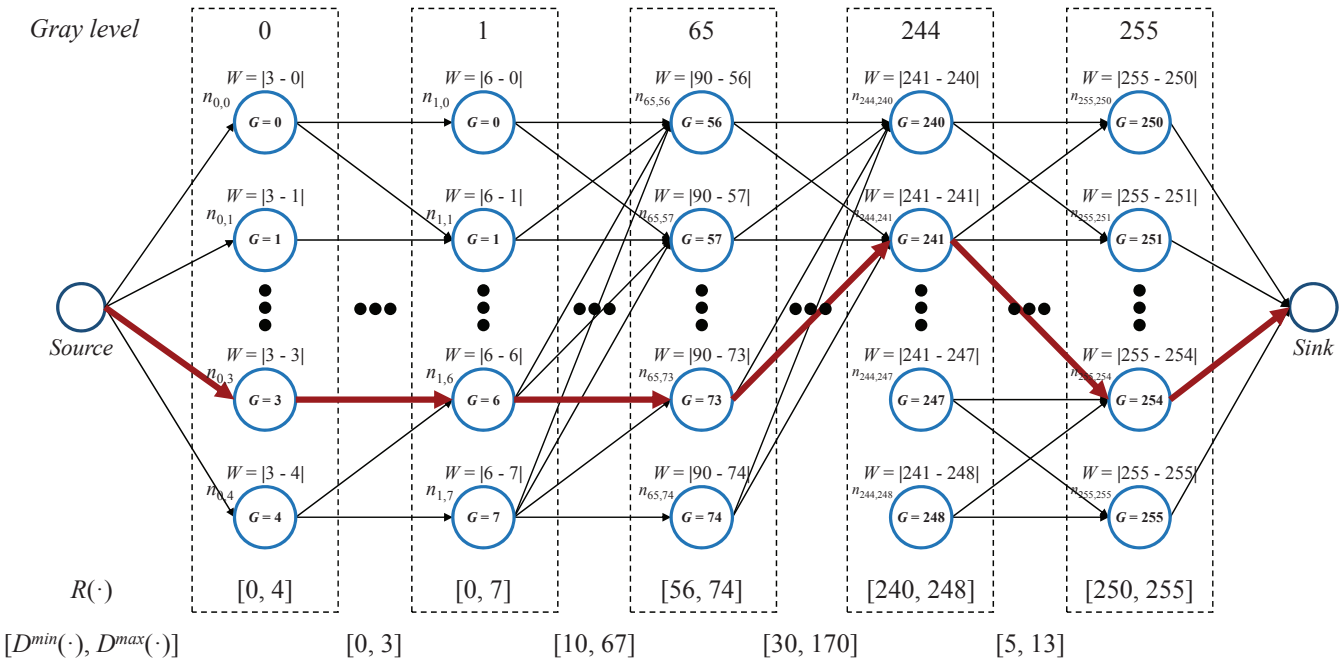


Figure 2.8: Illustration of formulating our histogram equalization problem using mapping ranges $R(\cdot)$ and mapping distances $[D^{min}(\cdot,\cdot), D^{max}(\cdot,\cdot)]$ into a network flow optimization problem.

[30]. Since the number of nodes in each column (i.e., $|R(\cdot)|$) is relatively very small and much less than L (total number of gray-levels), it can be bounded by a constant. Thus, the number of arcs between nodes in two adjacent columns is also bounded by a constant. Therefore, the time complexity of the shortest path algorithm is bounded by $O(|L|)$, which means that Step 2 is solvable in $O(L)$. Collectively, combined with the linear time computation of all $R(\cdot)$ and $[D^{min}(\cdot,\cdot), D^{max}(\cdot,\cdot)]$ in Step 1, the run time of HE-gc is $O(M+L)$ where M is the number of pixels in the input image.

2.4 Integration of Power Minimization

Power minimization is another important design objective in the histogram equalization. The power consumption, P_x, of an equalized image produced by applying a mapping function $x(\cdot)$ to an image I_0 is estimated as ([25],[31]):

$$P_x = \sum_{i=0}^{L-1} H(i) \cdot x(i)^{\gamma} \tag{2.14}$$

where $H(i)$ represents the intensity of gray-level i in I_0, $x(i)$ is the transformed gray-level of $i \in \{0, 1, \cdots, L-1\}$, L is the number of gray-levels in I_0, and γ is a correction parameter.

Consequently, the objective function to be minimized, updated from Δ_{diff} in $Eq.$ 2.13, is

$$\Delta_{diff+pwr} = \sum_{i=0}^{L-1} (|x(i) - x_0(i)| + \beta \cdot H(i) \cdot (x(i)^{\gamma} - x_0(i)^{\gamma})) \tag{2.15}$$

where β is a weighting factor and the value of γ varies depending on the display technology used. The only thing to do to incorporate the new objective function in $Eq.$ 2.14 is, in the network formulation, to replace the weight $w_{i,k} = |k - x_0(i)|$ assigned to each node $n_{i,k} \in V$ with $w_{i,k} = |k - x_0(i)| + \beta \cdot H(i) \cdot (k^{\gamma} - x_0(i)^{\gamma})$.

2.5 Experimental Results

We have tested our proposed histogram equalization algorithm HE-gc to evaluate the performance in various aspects. In the experiments, 24 images from the Kodak Lossless True Color Image suite are used as test images [32]. Additionally, we test 8 images in [33], which includes "girl", "F-16", "jelly beans" that have been popularly used in image processing.

Fig. 2.9 shows our 32 test images by grouped their mean brightness. The 24 Kodak images are partitioned into three grouped: *g1* for 8 darkest images, *g3* for 8 brightest images, and *g2* for the rest. The 8 popular images belong to *g4*.

To consider luminance component only, the color images in the test cases are converted to YUV color space. Then, we test the histograms constructed from the luminance component of Y. Since the test images are 8-bit pixel images, we set the gray-level range to $[0, 255]$, i.e., $L = 256$. In our experiment, we set the parameters of HE-gc as follow: $R_{UB} = 30$ (upper bound of mapping range), $\Lambda_g = 8$ (object's global property), $\Lambda_l = 2$ (object's local property), $\Lambda_d = 4$ (object's distance constraint), $\Sigma = 30$ (object's area constraint), and $D_{UB} = 4$ (upper bound of mapping distance). These parameters are set empirically.

We assess the effectiveness of HE-gc by measuring *image brightness*, *image quality* and *image contrast*.

1. To measure the brightness preservation, we use the AMBE (Absolute Mean Brightness Error) [11] metric, which is the absolute difference between the means of brightness of input and output images. The AMBE is computed by

$$AMBE = |\mu_x - \mu_y| \tag{2.16}$$

where μ_x is the mean gray-level of the pixels in the image, X, before the application of histogram equalization and μ_y is the mean gray-level in the image, Y, after the application.

Figure 2.9: Test images composed of 24 Kodak images and 8 popular images. The 24 Kodak images are partitioned into three grouped: *group1* for 8 darkest images, *group3* for 8 brightest images, and *group2* for the rest. The 8 popular images belong to *group4*.

The smaller the AMBE value is, the less the change of image brightness tends to be, AMBE measures the degree of luminance distortion of output image relative to the input image. One strength of HE-gc is that it is able to maintain a global brightness even under the power budget constraint since it well trade-offs between the power consumption and brightness by adjusting the weighting values of β in $Eq.\ 2.14$. We controlled the value of β in the objective function in $Eq.\ 2.15$ to produce images whose mean brightness is as close as to that of the input images.

2. To measure the image quality, we use the PSNR (Peak Signal to Noise Ratio), SSIM (Structural Similarity Index of Image) [34] and SW-SSIM (Structural similarity Weighted SSIM) [35] metrics. The SSIM is computed by

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_x^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.17}$$

where σ_{xy} is the covariance of gray-levels between the pixels in X and Y. σ_x and σ_y are the standard deviations of gray-levels in X and Y, respectively. C_1 and C_2 are positive constants. It is shown that $0 \leq SSIM(X,Y) \leq 1$.

The higher the PSNR value is, the better the image quality is, while the closer to 1 the values of SSIM and SW-SSIM are, the higher the image quality perceived by human visual is.

3. To evaluate the image contrast, we use EME [36] metric, which measures the block-level enhancement. EME is computed by

$$EME = \frac{1}{N_1 N_2} \sum_{l=1}^{N_2} \sum_{k=1}^{N_1} 20log \frac{g^w_{max;k,l}}{g^w_{min;k,l}} \tag{2.18}$$

where the $g^w_{max;k,l}$ and $g^w_{min;k,l}$ represent the lowest and highest gray-levels among the pixels in block $w_{k,l}$, respectively. $w_{k,l}$ is one of the blocks produced by partitioning an image by $N_1 \times N_2$.

The higher the value of EME is, meaning the larger the gap between $g^w_{max;k,l}$ and $g^w_{min;k,l}$, the higher the contrast of the sub-image corresponding to block $w_{k,l}$ is.

4. To increase the confidence of image evaluation, we perform a subjective assessment. The subjective assessment is conducted as follows: 20 viewers was responded to the image survey, consisting of 16 males and 4 females. The eight algorithms (i.e., HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc) were tested on the 24 Kodak images and the 8 popular images. The full tournament was adopted for the subjective assessment. In order to confirm the maintenance of natural look in the original image, the viewers check the original image in each tournament step. Total of 256 images are evaluated, and 896 tournament steps are performed.

5. We measure the run times of the algorithms. The experiment environment is 3.2GHz Intel i5 with 8.00 GB RAM. HE-gc and TE-HE are implemented with OpenCV version 2.4.9. Other algorithms are implemented by Matlab.

2.5.1 Evaluation of HE-gc for Image Quality, Contrast, and Brightness Preservation

HC-gc is applied to 24 Kodak images and 8 popular images, as shown in Fig. 2.10, to check the image quality, measured by PSNR, SSIM and SW-SSIM, image contrast, measured by EME while preserving the image brightness, measured by AMBE, together with the subjective assessment. We measured all the images produced by HE [3], which is the most naive method of gray-level partition based equalization, BBHE, which employs the bi-partition of gray-levels to [3], BPDFHE [14], which is also partition based equalization with additional emphasis on preserving image brightness, PCCE [25], which considers the trade-off between image quality and power consumption, NMHE [24], which places more importance on the contrast enhancement, TE-HE

Table 2.1: Comparison of subjective assessment for the images produced by HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc for 24 Kodak images and 8 popular images. The 24 Kodak images are partitioned into three grouped: *g1* for 8 darkest images, *g3* for 8 brightest images, and *g2* for the rest. The 8 popular images belong to *g4*.

Metric	Image	Ori.	HE [3]	BBHE [9]	BPDFHE [14]	PCCE [25]	NMHE [24]	TE-HE [27]	ESIHE [16]	HE-gc
Subjective assessment	g1	-	155	262	670	880	704	161	734	914
	g2	-	198	392	752	777	798	115	597	851
	g3	-	220	319	778	817	845	136	635	730
	g4	-	144	305	614	791	640	346	737	903
	Avg.	-	179	320	704	816	747	190	676	**850**

Table 2.2: Comparison of image quality (PSNR, SSIM, SW-SSIM) for the images produced by HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc for 24 Kodak images and 8 popular images. The 24 Kodak images are partitioned into three grouped: *g1* for 8 darkest images, *g3* for 8 brightest images, and *g2* for the rest. The 8 popular images belong to *g4*.

Metric	Image	Ori.	HE [3]	BBHE [9]	BPDFHE [14]	PCCE [25]	NMHE [24]	TE-HE [27]	ESIHE [16]	HE-gc
	g1	-	13.4	16.4	28.7	29.5	23.1	15.1	25.7	36.2
PSNR	g2	-	16.5	18.6	29.2	26.7	24.6	16.0	22.8	27.6
(Image quality)	g3	-	17.4	19.3	30.0	29.5	27.4	17.4	23.7	25.4
	g4	-	14.8	17.2	30.0	27.3	20.9	17.5	22.7	27.5
	Avg.	-	15.5	17.9	29.5	28.3	24.0	16.5	23.7	**29.2**
	g1	-	0.69	0.74	0.89	0.96	0.90	0.63	0.93	0.97
SSIM	g2	-	0.80	0.83	0.94	0.95	0.96	0.73	0.91	0.95
(Image quality)	g3	-	0.77	0.82	0.95	0.97	0.97	0.77	0.92	0.94
	g4	-	0.60	0.72	0.88	0.93	0.90	0.78	0.92	0.95
	Avg.	-	0.72	0.78	0.91	0.95	0.93	0.73	0.92	**0.95**
	g1	-	0.71	0.75	0.91	0.96	0.91	0.67	0.93	0.97
SW-SSIM	g2	-	0.79	0.83	0.95	0.95	0.95	0.73	0.92	0.95
(Image quality)	g3	-	0.80	0.84	0.96	0.97	0.97	0.76	0.91	0.94
	g4	-	0.61	0.72	0.92	0.92	0.88	0.74	0.91	0.93
	Avg.	-	0.73	0.79	0.94	0.95	0.93	0.72	0.92	**0.95**

Table 2.3: Comparison of image contrast for the images produced by HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc for 24 Kodak images and 8 popular images. The 24 Kodak images are partitioned into three grouped: *g1* for 8 darkest images, *g3* for 8 brightest images, and *g2* for the rest. The 8 popular images belong to *g4*.

Metric	Image	Ori.	HE [3]	BBHE [9]	BPDFHE [14]	PCCE [25]	NMHE [24]	TE-HE [27]	ESIHE [16]	HE-gc
	g1	11.6	15.4	15.6	12.7	13.6	13.3	17.2	13.9	13.4
EME	g2	10.0	14.7	14.5	12.6	12.9	11.3	16.3	13.7	13.2
(Image contrast)	g3	9.0	13.3	12.8	10.2	10.8	10.0	14.5	12.0	11.7
	g4	6.1	11.0	9.2	7.0	8.2	6.5	10.7	7.9	8.1
	Avg.	9.2	13.6	13.0	10.6	11.4	10.3	14.7	11.9	**11.6**

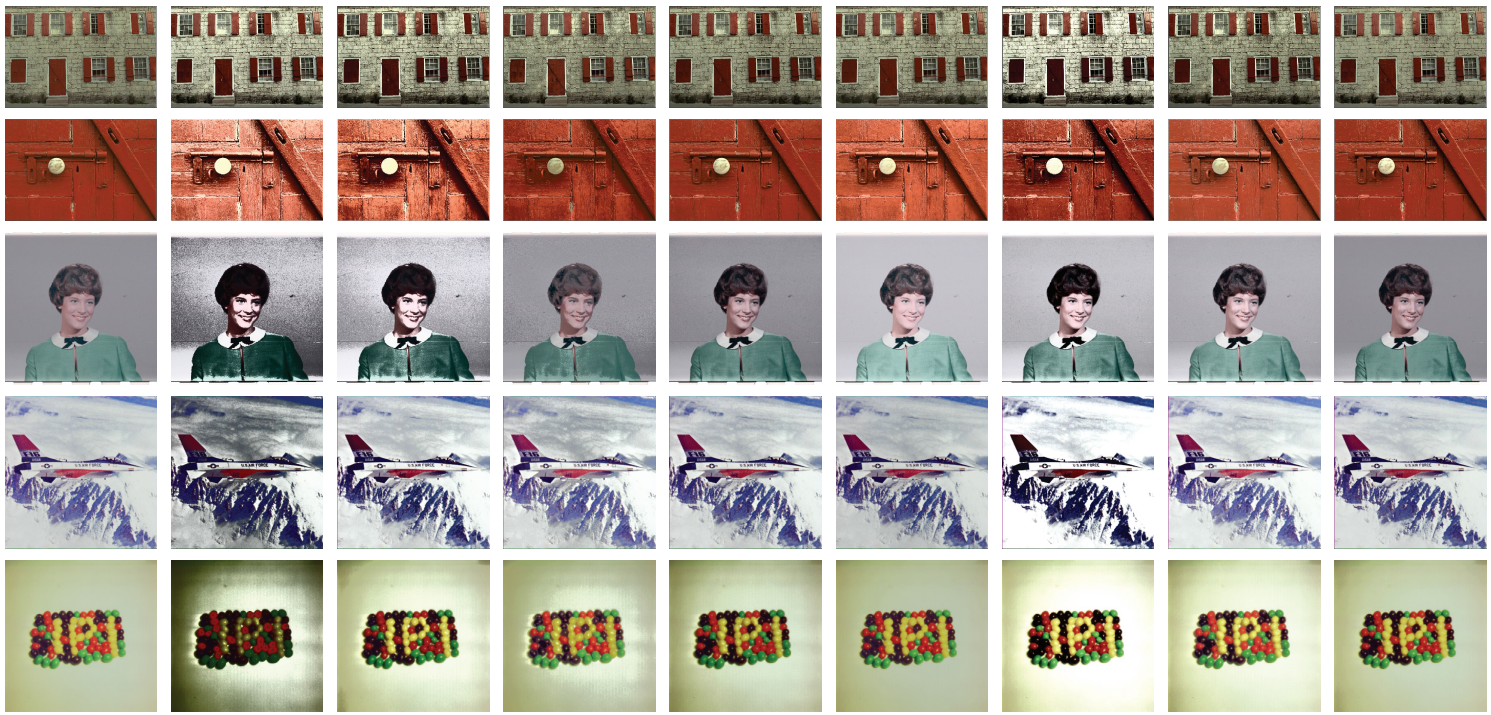
Table 2.4: Comparison of preservation of image brightness for the images produced by HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc for 24 Kodak images and 8 popular images. The 24 Kodak images are partitioned into three grouped: *g1* for 8 darkest images, *g3* for 8 brightest images, and *g2* for the rest. The 8 popular images belong to *g4*.

Metric	Image	Ori.	HE [3]	BBHE [9]	BPDFHE [14]	PCCE [25]	NMHE [24]	TE-HE [27]	ESIHE [16]	HE-gc
	g1	0	127.0	107.6	0.8	10.1	40.6	63.0	28.7	0.1
AMBE	g2	0	58.0	111.2	1.5	12.9	28.3	53.6	26.5	0.7
(Brightness change)	g3	0	46.5	131.3	1.3	8.7	17.5	30.5	19.3	4.7
	g4	0	69.1	156.9	0.6	11.5	63.9	30.5	41.7	0.6
	Avg.	0	75.1	126.7	1.0	10.8	37.6	51.9	28.8	**1.5**

Table 2.5: Comparison of excution time for the images produced by HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc for 24 Kodak images and 8 popular images. The 24 Kodak images are partitioned into three grouped: *g1* for 8 darkest images, *g3* for 8 brightest images, and *g2* for the rest. The 8 popular images belong to *g4*.

Metric	Image	Ori.	HE [3]	BBHE [9]	BPDFHE [14]	PCCE [25]	NMHE [24]	TE-HE [27]	ESIHE [16]	HE-gc
Execution time (ms)	g1	-	1.0	34.4	9.0	171.6	17.8	894.8	51.1	70.8 (5.0)
	g2	-	1.0	34.5	9.0	218.9	17.1	932.3	57.3	69.0 (5.0)
	g3	-	1.0	33.9	9.0	137.9	15.3	905.6	52.1	70.5 (5.3)
	g4	-	0.6	13.5	4.9	324.5	5.3	431.8	18.4	31.5 (3.9)
	Avg.	-	0.9	29.1	8.0	213.2	13.8	791.1	44.7	**60.4 (4.8)**

Figure 2.10: Result images processed by our HE-gc method. The 24 Kodak images are partitioned into three grouped: *group1* for 8 darkest images, *group3* for 8 brightest images, and *group2* for the rest. The 8 popular images belong to *group4*.



(*a*) Original (*b*) HE [3] (*c*) BBHE [9] (*d*) BPDFHE [14] (*e*) PCCE [25] (*f*) NMHE [24] (*g*) TE-HE [27] (*h*) ESIHE [16] (*i*) HE-gc

Figure 2.11: Comparison of the images produced by the conventional histogram equalization methods HE [3], BBHE [9], BPDFHE [14], PCCE [25], NMHE [24], TE-HE [27], ESIHE [16], and our HE-gc for the two Kodak images and "girl", "F-16", "jelly beans" images.

[27] which uses TV-L^a (where, $a = 1$) for texture enhancement, ESIHE [16] which is a recent partition based equalization, and our HE-gc.

Table 2.1, Table 2.2, Table 2.3, Table 2.4, and Table 2.5 summarize the values of AMBE, EME, PSNR, SSIM, and SW-SSIM for the original images and the images produced by the conventional equalization methods HE, BBHE, BPDFHE, PCCE, NMHE, TEHE, ESIHE, and the images produced by our HE-gc. The 24 Kodak test images are grouped into three parts: *g1* for the 8 images with the lowest brightness, *g3* for the 8 images with the highest brightness, and *g2* for the rest. The 8 popular images belong to *g4*. The results show that (1) in terms of image quality, our PSNR value (= 29.2) is the higher, and SSIM and SW-SSIM values (= 0.95 and 0.95) is close to 1; (2) in terms of image contrast, our EME value (= 11.6) is moderate; (3) in terms of preserving image brightness, ours is excellent and comparable to that in BPDFHE whose primary concern is preserving brightness.

Since HE-gc has an enough degree of contrast enhancement (EME), it is observed little difference with respect to the original image (PSNR), and similar image quality based on a human visual system with respect to the original image (SSIM/SW-SSIM). That is, our goal of contrast enhancement with the maintenance of naturalness of the original image is achieved. Furthermore, the subjective assessment value of 850 validates the achievement.

Overall, our HE-gc produces images with better image quality while almost strictly preserving image brightness as that of the original images, achieving a nontrivial improvement on image contrast.

Fig. 2.11 shows the image comparison for two Kodak test cases and “girl”, “F-16”, “jelly beans” test cases. It is shown a noticeable improvement on visual quality by HE-gc, in particular the “girl” image, is achieved. For “girl” image, there is no background noise with contrast enhancement in HE-gc while the other methods cause background noises. This is because our method considers the background as a cluster, so that the

contrast of the inside of the cluster is maintained.

Fig. 2.12, Fig. 2.13, Fig. 2.14, Fig. 2.15, and Fig. 2.16 respectively show the comparison of the result images of algorithms BPDFHE, PCCE, NMHE, TEHE, and ESIHE with our proposed algorithms HE-gc. By comparing the results of other algorithms, we can confirm that our proposed algorithms prevents over-enhancement and has better contrast enhancement much more than other algorithms. In particular, our proposed algorithm has many strengths in reducing distortion of noise and luminance in the background image and in enhancing the contrast of features.

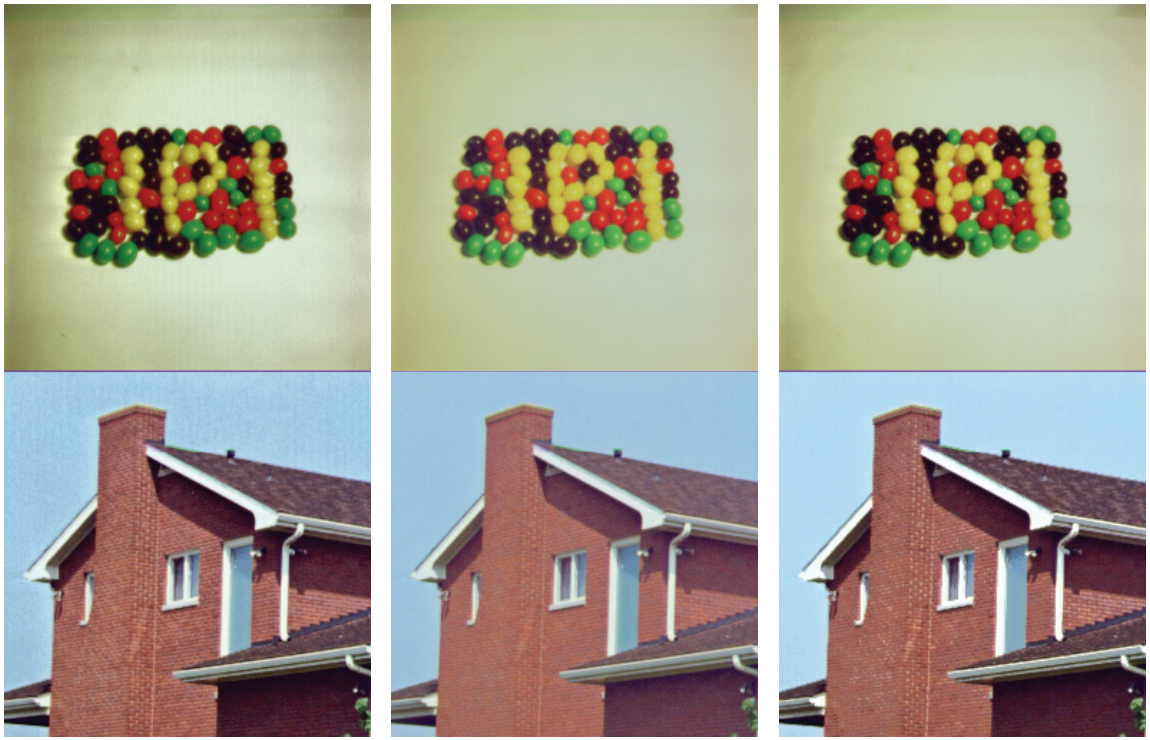
Regarding run time, the value in the parentheses in HE-gc represents the run time spent by the execution of the constructed network in HE-gc. Since the network processing can be parallelized by multi-processing environment, the time could be greatly reduced.

2.5.2 Evaluation of HE-gc for Power Consumption

Since to the best of our knowledge, PCCE [25] is the only work which has addressed the power minimization in the course of histogram equalization, we compare the results produced by HE-gc with that by PCCE. We applied HE-gc multiple times to the original images, increasing each time the value of the power weighting factor β in the objective function in $Eq.\ 2.15$ to find images whose mean brightness is as close as to that of the images produced by PCCE. Fig. 2.17 shows the differences of image contrast, measured by EME metric, image quality, measured by PSNR and SSIM metrics, and power consumption, measured by the formulation in $Eq.\ 2.14$ with the exponent parameter $\gamma = 2.2$ between the images produced by PCCE and HE-gc for 24 Kodak test images. The results are normalized with respect to the measurements of the images produced by PCCE. As shown in Fig. 2.17(a), the image contrast by HE-gc is a little worse, but as shown in Figs. 2.17(b) and 2.17(c), the image quality by HE-gc is superior to that by PCCE. In terms of power consumption as indicated in Fig. 2.17(d),

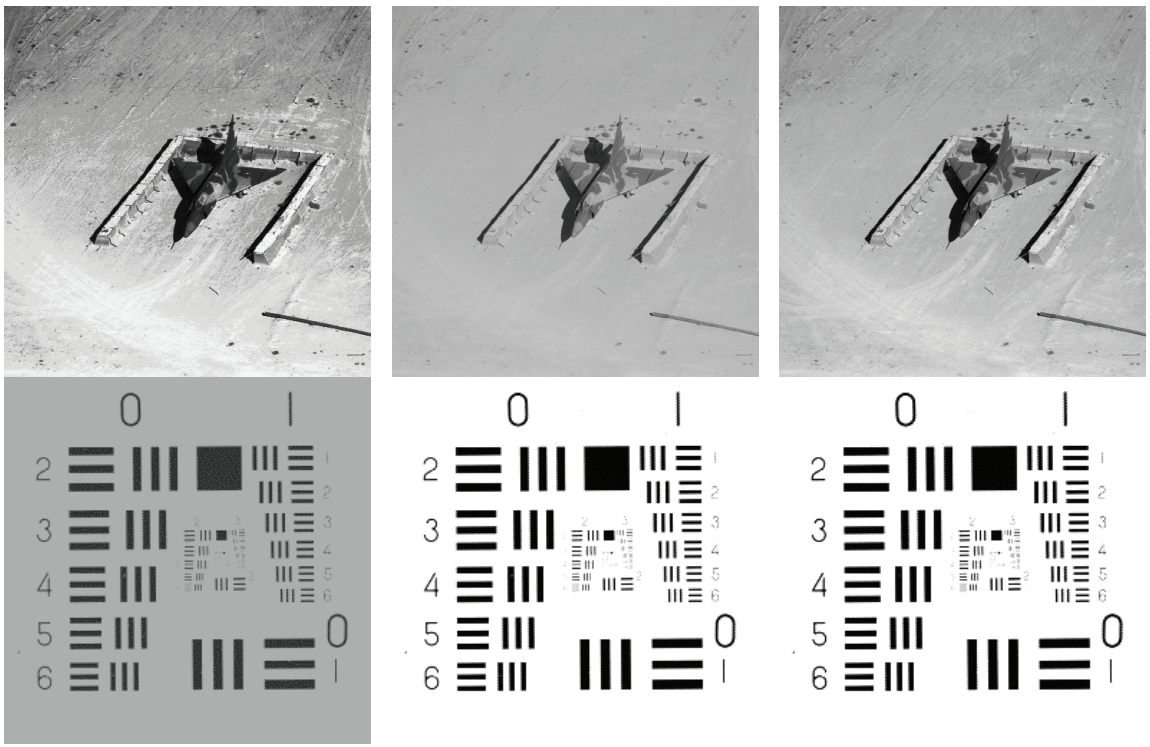
(*a*) BPDFHE [14] **(*b*) Original** **(*c*) HE-gc**

Figure 2.12: Comparison of the images produced by BPDFHE [14] and our HE-gc for “girl” and “F-16” images.



(*a*) **PCCE [25]** (*b*) **Original** (*c*) **HE-gc**

Figure 2.13: Comparison of the images produced by PCCE [25] and our HE-gc for “jelly beans” and “house” images.



(*a*) NMHE [24] (*b*) Original (*c*) HE-gc

Figure 2.14: Comparison of the images produced by NMHE [24] and our HE-gc for “airplane” and “resolution chart” images.

(*a*) TEHE [27] **(*b*) Original** **(*c*) HE-gc**

Figure 2.15: Comparison of the images produced by TEHE [27] and our HE-gc for “mandrill” and “splash” images.

(*a*) ESIHE [16] **(*b*) Original** **(*c*) HE-gc**

Figure 2.16: Comparison of the images produced by ESIHE [16] and our HE-gc for two Kodak images.

HE-gc is outstanding in saving power. The images marked with small dark dots on the curves in Fig. 2.17 are shown in Fig. 2.18. Finally, Table 2.6 shows the average values of the measurement of EME, PSNR, SSIM, and power saving (P_{save}) of the images produced by PCCE [25] and HE-gc for 24 Kodak images. In summary, it is confirmed that HE-gc is able to reduce the power consumption by 22.3% further over the power saving by PCCE [25] while comparably maintaining the image quality and contrast.

Table 2.6: Comparison of subjective assessment, image quality (PSNR, SSIM), image contrast (EME), and power consumption (P_{save}) of the images produced by PCCE [25] and our HE-gc for 24 Kodak images. The averaged results are summarized.

	PCCE [25]				HE-gc			
	EME	*PSNR*	*SSIM*	$P_{save}(\%)$	*EME*	*PSNR*	*SSIM*	$P_{save}(\%)$
Avg.	12.7	23.8	0.940	14.7	12.4	24.8	0.952	17.9
($\Delta\%$)	-	-	-	-	**-2.5**%	**3.8**%	**1.2**%	**22.3**%

2.6 Summary

In this chapter, we proposed a new histogram equalization method to overcome the limitation of the previous gray-level partition based histogram equalizations, which were not able to equalize a sub-histogram crossing a partition boundary. Precisely we devised two enabling techniques called *mapping range* and *mapping distance* to make a full exploitation of mapping flexibility of gray-levels. With the integration of the two enabling techniques, we formulated the histogram equalization problem into a flow optimization problem in a network and solved it globally and efficiently. In addition, it was shown that the factor of minimizing power consumption was seamlessly combined to our network flow optimization framework. Through experiments with the diverse images, it was demonstrated that the proposed histogram equalization

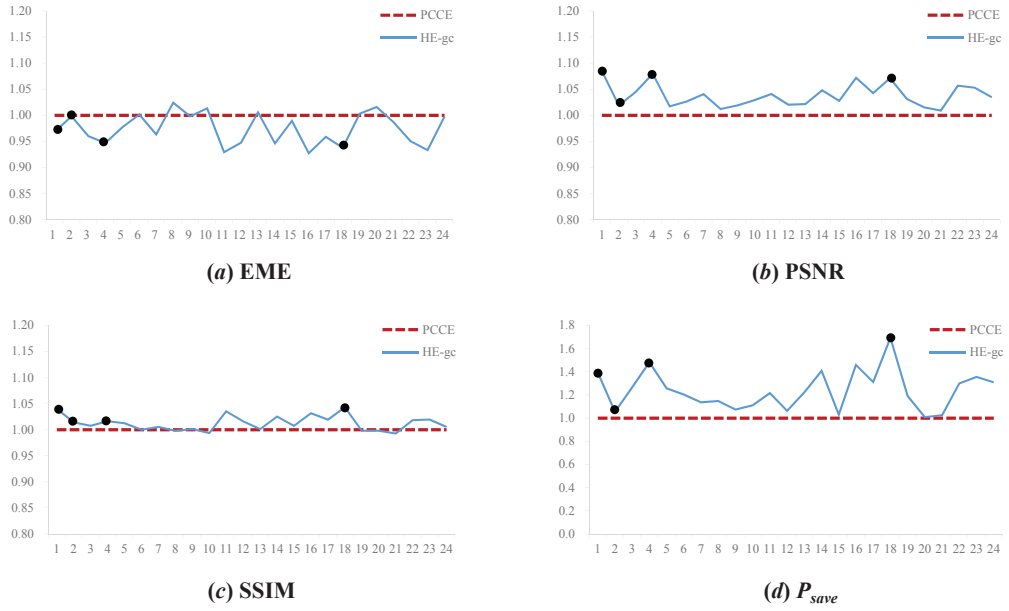
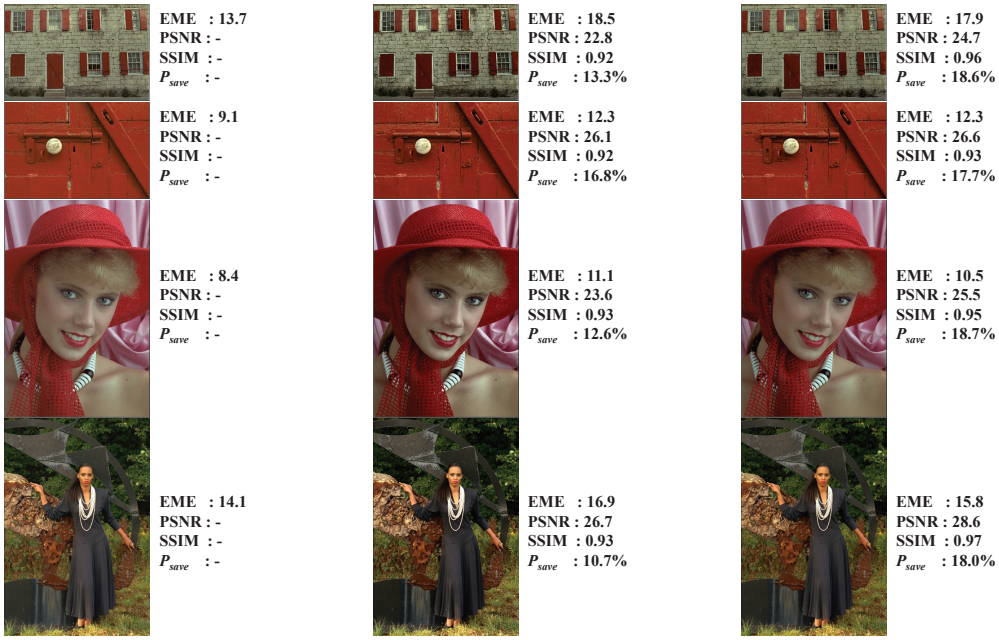


Figure 2.17: Comparison of contrast enhancement (EME), image quality (PSNR, SSIM), and power saving (P_{save}) for the images produced by PCCE [25] and HE-gc for 24 Kodak images. The horizontal axis enumerates the 24 input images. The HE-gc results are normalized to that of PCCE. The curves show that the image contrast by HC-gc is a little worse, but the image quality is consistently better than PCCE, and power is prominently saved by HE-gc.



(*a*) **Original** (*b*) **PCCE [25]** (*c*) **HE-gc**

Figure 2.18: Comparison of the images produced by PCCE [25] and our HE-gc for the 4 Kodak images.

was a viable solution to improving image quality consistently even saving power for a broad spectrum of images. The implementation of histogram equalization without over-enhancement through the network flow method used in the proposed algorithm is also an important contribution to the image processing area.

Chapter 3

Community-Adaptive Link Prediction in Social Networks

3.1 Introduction

In recent big data era, social network analysis becomes an important research theme. In particular, the link prediction problem in the social network analysis, which tries to find the missing links or predict the future links, is one of the hot research topics [4, 5]. There is a variety of applications of link prediction, such as partner or friend recommendation system and collaboration prediction system among researchers in academic. There are several approaches that have been applied in the domain of link prediction problem. The approaches can be grouped into a number of significant types such as content-based models, probabilistic models, and similarity-based models [6]. Among those, the similarity-based models, providing high prediction accuracy, are the most popular approaches [7, 8], in which a topological similarity measure is applied to every pair of nodes with missing a link to forecast the likelihood of the occurrence of a link in the future. The similarity-based models produce similarity scores, based on such measures, for the node pairs, and use them to perform the link prediction task.

To enhance the prediction accuracy, many similarity-based models have attempted to integrate the community information, if available, in the social network structure

into their link score formulations. Soundarajan and Hopcroft [39] extended the score formulation of two nodes employed in the common neighbor [37] and resource allocation [38] methods by including an 'adding' term to strengthen the score contribution by the common neighbors in the same community as that of the nodes. Precisely, if the amount of score contributed by a common neighbor is C in [37, 38], their updated score formulation doubles its contribution (i.e., $2C$) if the common neighbor is in the same community. Similarly, Valverde-Rebaza and Lopes [41] extended the score formulation of two nodes used in the common neighbor method [37] by including a 'dividing' term to weaken the score contribution by the common neighbors in different communities from that of the nodes. That is, if the score of a pair of nodes contributed by all their common neighbors is S, and among them, K nodes are not in the same community as that of the pair of nodes, they modify the score to S/K. Recently, Ding *et al.* [44] scored link purely based on the community information. They generate a set of multi-resolution communities by applying a community detection algorithm multiple times by varying modularity (i.e., community formation) density function, and place higher (or lower) scores for node pairs that are in the same community in more (less) multi-resolution communities.

One common strategy of the prior community-aware link prediction algorithms is that they devised a sort of unified link prediction formulation that simply includes a premium term to express whether a link is structurally in the same community or not. However, since the formulation of the premium term relies on the structural formation of communities only, it cannot take into account the fact that the communities in different social networks, though they form almost identical community structures, can make different levels of influence on the link prediction. To cope with this limitation, we propose a community-adaptable approach, in which we use two separate link predictions depending on inter or intra-links in a community, and then balance the links based on the degree of community influence on the link prediction.

3.2 Related Works

3.2.1 Neighbor-Based Link Prediction

The neighbor-based method exploits the information of common neighbors shared by two nodes v_i and v_j.

(1) *Common neighbor method (CN)*: The common neighbor method (CN) [37] proposed by Newman is the most famous link prediction algorithm. The score of two nodes v_i and v_j is rated the number of common neighbors. The definition of CN is as follows:

$$CN(i,j) = |\Gamma(i) \cap \Gamma(j)| \tag{3.1}$$

where the $\Gamma(i)$ is the neighbors of the vertex v_i.

(2) *Resource allocation method (RA)*: Resource allocation method (RA) [38] is proposed by Zhou *et al.*. It scores the node pair v_i and v_j by the summation of inversed degrees of common neighbors. The definition of RA is as follows:

$$RA(i,j) = \sum_{k \in \Gamma(i) \cap \Gamma(j)} \frac{1}{d(k)} \tag{3.2}$$

where the $d(k)$ is the degree of the vertex v_k.

3.2.2 Community-Aware Link Prediction

This section describes a set of representative link prediction works that have considered the community information in their similarity scoring formulation.

(1) *Measuring community information of individual common neighbors*: Soundarajan and Hopcroft [39] used InfoMap [40] community detection algorithms to find community information. InfoMap identifies communities by minimizing the length of random

walks used in the information theory. Then, they extended the CN (common neighbor) [37] and RA (resource allocation) [38] based similarity algorithms by adding the community information to the link scoring formulations. The updated scorings, called $CNIM(i,j)$ and $RAIM(i,j)$, between two nodes v_i and v_j are defined as:

$$CNIM(i,j) = |\Gamma(i) \cap \Gamma(j)| + \sum_{k \in \Gamma(i) \cap \Gamma(j)} C(i,j,k) \tag{3.3}$$

$$RAIM(i,j) = \sum_{k \in \Gamma(i) \cap \Gamma(j)} \frac{1 + C(i,j,k)}{d(k)} \tag{3.4}$$

where $\Gamma(i)$ and $\Gamma(j)$ represent the sets of neighbors of vertices v_i and v_j, respectively. $C(i,j,k)$ = 1 if vertices v_i, v_j, and v_k belong to the same community, and $C(i,j,k)$ = 0, otherwise. $d(k)$ indicates the degree of vertex v_k.

(2) *Measuring the community relation among common neighbors*: Valverde-Rebaza and Lopes [41] used the label propagation based community detection algorithm in [43] and proposed a link prediction by devising a measure, so-called within and inter community measure (WIC) [42]:

$$WIC(i,j) = \begin{cases} |\Lambda^W(i,j)| & \text{if } \Lambda^W_{i,j} = \Gamma(i) \cap \Gamma(j) \\ \frac{|\Lambda^W(i,j)|}{|\Lambda^I(i,j)|} & \text{otherwise} \end{cases} \tag{3.5}$$

in which $\Lambda^W(i,j)$ and $\Lambda^I(i,j)$ indicate the set of within-community common neighbors of v_i and v_j, i.e., $C(i,j,-)$ = 1 and the set of inter-community common neighbors, i.e., $C(i,j,-)$ = 0, respectively.

(3) *Measuring based on community information from multi-resolution community division [44]*: Ding *et al.* [44] proposed the link prediction measure called *CommPre* (CP) that fully incorporates multi-resolution communities information. They repeatedly applied the community detection algorithm in [46] with the modularity density function in [45] to produce the multi-resolution community division. Precisely, they applied the

detection algorithm M times, i.e., $m = 1, \cdots, M$, by varying the community resolution from $\lambda_1, \cdots,$ to λ_M. The measure $CP(i, j)$ between vertices v_i and v_j is defined as:

$$CP(i,j) = \sum_{m=1}^{M} \alpha_m C_m(i,j) \tag{3.6}$$

in which α_m is the number of communities produced when the community resolution is set to λ_m, and $C_m(i, j)$ = 1 if v_i and v_j belongs to the same community among the communities produced when λ_m is used as the community resolution and $C_m(i, j)$ = 1, otherwise.

3.3 Algorithm for Community-Adaptive Link Prediction

The inputs to our community-adaptive link prediction algorithm are (1) the network of nodes, $G(V, E)$, in a directed graph form to predict new links and (2) the link evolvement, $\mathcal{E}$ on the network in the prior time stamp. Thus, our algorithm will make a more accurate link prediction on $G(V, E)$ by exploiting $\mathcal{E}$ and community information. More precisely, rather than computing the similarity score between nodes v_i and v_j in a single equation like $CNIM(i, j)$, $RAIM(i, j)$, and $WIC(i, j)$ in [39, 42], we divide the missing links in G into two types: *intra-links* which indicate the links whose ending nodes are in the same community and *inter-link* for the rest, and compute two sets of similarity scores, one for intra-links and another for inter-links. Thus, at this step, we do not have to worry about integrating the community contribution into the score calculation. Then, we find a value of λ that balances the two sets of scores and that leads to a full reflection of the prior link evolvement $\mathcal{E}$. We call λ the balance factor for G and $\mathcal{E}$. Note that to get the data of $\mathcal{E}$ as input. we analyzed that the network of the previous time stamp by dropping 10% edges randomly. Fig. 3.1 shows the four steps of our link prediction.

1. *Extracting community information on* G: To classify the missing links in G into intra and inter-links, it is required to extract the community information on G. We simply used the multi-resolution community division in [44], in which we set up the modularity density to the maximum level.

2. *Generating intra and inter-links*: This step is straightforward, simply dividing the missing links into two sets S_{intra} for intra-links and S_{inter} for inter-links. See Fig. 3.2 for illustration.

3. *Computing similarity scores for missing links*: We can apply any link prediction algorithm to the links in S_{intra} and S_{inter}. In this work, we choose the Common Neighbor method (CN) [37] and Resource Allocations method (RA) [38]. The following equations labeled $CACN(i,j)$ and $CARA(i,j)$ describe our similary scores for a link between nodes v_i and v_j. Note that scores are weighted by the balance factor λ for the intra-links and by 1 - λ for the inter-links.

$$CACN(i,j) = \begin{cases} \lambda CN(i,j) & \text{, if link}(i,j) \in S_{intra} \\ (1-\lambda)CN(i,j) & \text{, if link}(i,j) \in S_{inter} \end{cases} \tag{3.7}$$

$$CARA(i,j) = \begin{cases} \lambda RA(i,j) & \text{, if link}(i,j) \in S_{intra} \\ (1-\lambda)RA(i,j) & \text{, if link}(i,j) \in S_{inter} \end{cases} \tag{3.8}$$

4. *Determining the breakeven value of* λ: The breakeven value of λ is the score weight which will make a maximal link prediction if it were applied to $\mathcal{E}$. Thus, by examining the link evolvement in $\mathcal{E}$, we can find the breakeven value of λ. The weighted scores are then used to predict links that are most likely to appear in the future.

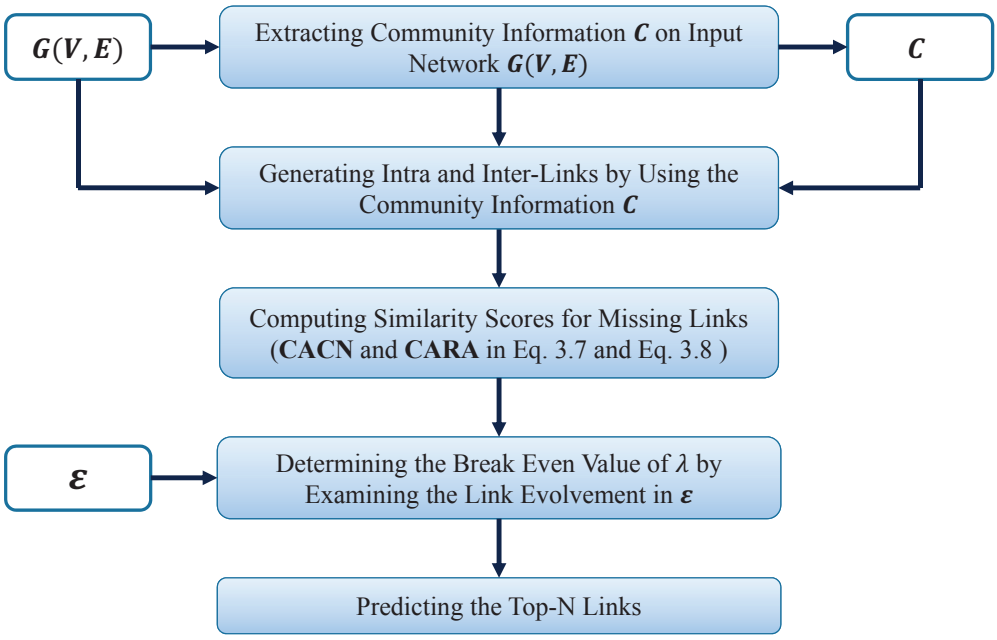


Figure 3.1: Alogrithm flow of our community-adaptive link prediction algorithm.

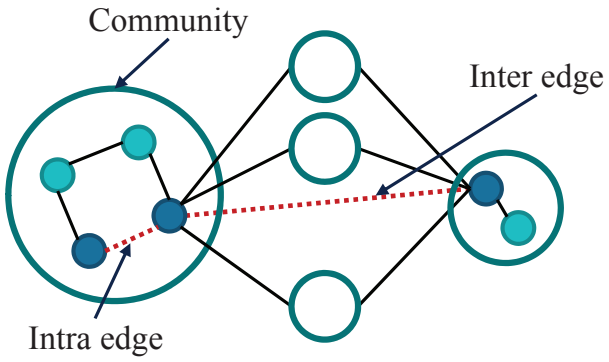


Figure 3.2: Separation of the links by the community.

3.4 Experimental Results

3.4.1 Experimental Setups

1. **Environment**: As our experiment environment, we have tested 3.2GHz Intel i5 with 8.00 random access memory. All of the link prediction algorithms were implemented with C++ language.

2. **Data Sets**: In order to evaluate the algorithm, we have used several famous data sets in the social network area. In the experiments, all datasets are considered as undirected and unweighted network. Table 3.1 shows a summary of the six networks used in our test. Advogato, Facebook, and Hamster of social types are basic networks composed of connection between friends. In the citation types of network DBLP, nodes are documents with the connection between referenced each other. Besides that, network GRQC and network WikiVote include data of collaboration and voting administrator, respectively.

3. **Comparison Algorithms**: As the comparison algorithms, we have tested the neighbor-based, community-aware link prediction algorithms, and our community-adaptive prediction algorithms. In the neighbor-based method, the methods of CN and RA are tested. In the community-aware methods, the methods of CNIM, RAIM, WIC, and CP are tested. CNIM and RAIM exploit the Infomap community detection algorithms to obtain the community information. WIC and CP use the label propagation and modularity density methods, respectively as the community detection algorithms. In WIC, lambda parameter controlling the resolution of community division is set to the values in range of [0, 1] with 0.1 increment. Lastly, we have also tested our two community-adaptive methods: CACN and CARA. To find the best precision, balance parameter λ is set to the values from 0 to 1 with 0.1 increment

Table 3.1: Summary of six experimental networks Advogato, DBLP, Email, Facebook, Hamster, and WikiVote. The V and E are the size of vertices and edegs, respectively.

Networks	Descriptions	$\|V\|$	$\|E\|$
Advogato [47]	Social network for containing relationship links of the online community Advogato.	6541	51127
DBLP [48]	Citation network of publications in DBLP.	12591	49743
Facebook [49]	Social network for containing friendship links of the Facebook.	4039	88324
GRQC [50]	Collaboration network of the General relativity and Quantum Cosmology in arXiv.	5242	14496
Hamster [51]	Social network for containing friendship links of the website hamsterster.com.	2426	16631
WikiVote [52]	Network of users from the English Wikipedia that voted for admin elections.	7118	103675

4. **Evaluation**: Since all test datasets are static data, we used two methods to separate a training set and a test set in the experiments. First, 10-fold cross validation, which is one of the well-known methods, is used to make a training network. This method divides the given network into 10 partitions and uses the 9 pieces of partitions as the training set while using a single piece of partition as the test set. Thus, 90% of the network is used in the link prediction methods with all of the node pairs scoring, and then 10% of the network is used to evaluate the performance of each link prediction method. Since we use each piece of partition to compose each different test set, we suggested the average value of the predicted results after 10 times iterations for testing.

   Second, to evaluate the link prediction performance in each different level of community influence, a specially designed Bernoulli trial, in which probability of choosing an intra-link to the test set is p, is exploited to delete the 10% edges to organize a test set. Therefore, if an edge is selected for a test set, the probability of being an intra-link is p while an inter-link is $1 - p$. To create a test set that can evaluate a network which has identical community structures and different level of community influence, we composed several test sets with different probability of choosing intra-links. In the experiments, a probability of choosing intra-links is set to 0.05, 0.1, and 0.5.

   To evaluate the algorithm performance, *Top-N precision* is used in our experiments. Top-N precision is the precision of top N node pairs in a sorted list as decreasing order after scoring in link prediction. Let N_i be the number of links in the training set as the answer within the N node pairs selected by link prediction method. Then, Top-N precision is $\frac{N_i}{N}$. we have used N = 300 in our experiments.

Table 3.2: Comparison of Top-300 precision for link prediction processed by neighborbaed methods (CN and RA), community-aware methods (CNIM, RAIM, WIC, and CP), and community-adaptive methods (CACN and CARA) for networks Advogato, DBLP, Facebook, GRQC, Hamster, and WikiVote with 10-fold cross validation as a test set type.

Networks	CN	RA	CNIM	RAIM	WIC	CP	CACN(λ)	CARA(λ)
Advogato	0.275	0.286	0.378	0.447	0.275	0.081	0.415(0.8)	0.563(1)
DBLP	0.133	0.143	0.144	0.125	0.125	0	0.14(0.2)	0.132(0.2)
Facebook	0.916	0.922	0.907	0.906	0.864	0.223	0.902(1)	0.91(1)
GRQC	0.984	0.987	0.984	0.988	0.983	0.538	0.988(1)	0.995(1)
Hamster	0.363	0.433	0.409	0.794	0.305	0.648	0.714(1)	0.853(0.9)
WikiVote	0.244	0.231	0.256	0.195	0.237	0.005	0.253(0.1)	0.193(0.1)
Average	0.486	0.5	0.513	0.576	0.465	0.249	0.569	0.608

Table 3.3: Comparison of Top-300 precision for link prediction processed by neighborbaed methods (CN and RA), community-aware methods (CNIM, RAIM, WIC, and CP), and community-adaptive methods (CACN and CARA) for networks Advogato, DBLP, Facebook, GRQC, Hamster, and WikiVote with $p = 0.05$ of Bernoulli trial as a test set type.

Networks	CN	RA	CNIM	RAIM	WIC	CP	CACN(λ)	CARA(λ)
Advogato	0.28	0.3	0.333	0.39	0.277	0.002	0.343(0.4)	0.383(0.4)
DBLP	0.147	0.12	0.15	0.123	0.13	0	0.14(0)	0.123(0)
Facebook	0.463	0.883	0.397	0.76	0.253	0.033	0.737(0)	0.95(0)
GRQC	0.633	0.663	0.607	0.59	0.623	0.13	0.677(0.2)	0.793(0.2)
Hamster	0.317	0.477	0.387	0.423	0.263	0.113	0.303(0.5)	0.437(0.4)
WikiVote	0.243	0.183	0.243	0.186	0.243	0	0.25(0.1)	0.167(0.3)
Average	0.347	0.438	0.353	0.412	0.298	0.049	0.408	0.476

Table 3.4: Comparison of Top-300 precision for link prediction processed by neighborbaed methods (CN and RA), community-aware methods (CNIM, RAIM, WIC, and CP), and community-adaptive methods (CACN and CARA) for networks Advogato, DBLP, Facebook, GRQC, Hamster, and WikiVote with $p = 0.1$ of Bernoulli trial as a test set type.

Networks	CN	RA	CNIM	RAIM	WIC	CP	CACN(λ)	CARA(λ)
Advogato	0.28	0.303	0.35	0.427	0.277	0.001	0.373(0.7)	0.447(0.7)
DBLP	0.13	0.117	0.13	0.117	0.13	0	0.123(0.1)	0.12(0.2)
Facebook	0.55	0.90	0.517	0.797	0.343	0.009	0.67(0.5)	0.94(0)
GRQC	0.64	0.837	0.637	0.797	0.64	0.277	0.77(0.1)	0.853(0.6)
Hamster	0.29	0.523	0.423	0.513	0.26	0.143	0.42(0.6)	0.53(0.5)
WikiVote	0.25	0.20	0.263	0.18	0.25	0	0.25(0.1)	0.22(0.2)
Average	0.357	0.48	0.387	0.472	0.317	0.087	0.434	0.518

Table 3.5: Comparison of Top-300 precision for link prediction processed by neighborbaed methods (CN and RA), community-aware methods (CNIM, RAIM, WIC, and CP), and community-adaptive methods (CACN and CARA) for networks Advogato, DBLP, Facebook, GRQC, Hamster, and WikiVote with $p = 0.5$ of Bernoulli trial as a test set type.

Networks	CN	RA	CNIM	RAIM	WIC	CP	CACN(λ)	CARA(λ)
Advogato	0.2	0.293	0.32	0.433	0.203	0.123	0.5(1)	0.633(1)
DBLP	0.107	0.123	0.147	0.126	0.113	0	0.147(0.4)	0.15(0.3)
Facebook	0.923	0.93	0.917	0.907	0.847	0.207	0.827(0.4)	0.897(0)
GRQC	0.983	0.973	0.983	0.973	0.98.	0.693	0.983(1)	0.983(1)
Hamster	0.36	0.717	0.377	0.817	0.307	0.51	0.797(0.8)	0.92(0.9)
WikiVote	0.173	0.143	0.203	0.167	0.173	0.033	0.207(0.3)	0.16(0.2)
Average	0.458	0.53	0.491	0.571	0.438	0.261	0.577	0.624

Table 3.6: Comparison of average Top-300 precision for link prediction processed by neighbor-baed methods (CN and RA), community-aware methods (CNIM, RAIM, WIC, and CP), and community-adaptive methods (CACN and CARA) for networks Advogato, DBLP, Facebook, GRQC, Hamster, and WikiVote with 10-fold cross validation and $p = 0.05$, $p = 0.1$, $p = 0.5$ of Bernoulli trial as the test set types.

Types	CN	RA	CNIM	RAIM	WIC	CP	CACN	CARA
10-fold	0.486	0.5	0.513	0.576	0.465	0.249	0.569	0.608
p=0.05	0.347	0.438	0.353	0.412	0.298	0.049	0.408	0.476
p=0.1	0.357	0.48	0.387	0.472	0.317	0.087	0.434	0.518
p=0.5	0.458	0.53	0.491	0.571	0.438	0.261	0.577	0.624
Average	0.412	0.487	0.436	0.508	0.379	0.162	0.497	0.556

3.4.2 Evaluation of Top-N precision

Table 3.2 shows the result of Top-300 precision when 10-fold cross validation is used to get the test set. Table 3.3, Table 3.4, and Table 3.5 displays the result of Top-300 precision when the possibility of selecting intra-link (p) is respectively 0.05, 0.1, and 0.5. Since the CARA method shows the best performance in every result in Table 3.2, Table 3.3, Table 3.4, and Table 3.3, we can confirm that the proposed algorithms reflect the community influence of different levels in each network adaptively.

Table 3.6 summarizes the results of average Top-300 precision processed by existing and proposed methods. In the neighbor-based link methods, CN and RA are applied to the networks for evaluation of precision. RA and CA as reference methods show the average precision of 0.412 and 0.487, respectively.

In the community-aware methods, CNIM, RAIM, WIC, and CP are tested for evaluation. The precision of CNIM and RAIM are enhanced about 5% compared with their reference methods CN and RA. WIC and CP show poor precision values because they

only focus the community information without considering the common neighbor information.

In Table 3.6, our proposed methods are compared for evaluation of Top-300 precision. Table shows that the precision value of our $CARA$ is best in compared with the results by other comparison algorithms in Top-300 precision. The average value of CARA with highest Top-300 precision values (= 0.556) enhanced by about 15% over that by RA reference method.

3.5 Summary

In this chapter, we proposed a new community-adaptable link prediction method to overcome the limitation of the previous community-aware methods, in which the similarity scoring was not tightly taken into account to the prior information of the community influence on link generation. To reflect the degree of the community influence, we separate link similarity scorings, depending on whether the links are inter or intra-links under community scope. Then, by utilizing the prior data on the link evolvement in the network, we were able to find a breakeven balancing weight between the scores of inter and inter-links to predict the top future links from all the missing links.

Chapter 4

Conclusion

The contributions of our proposed methods, gray-level context-driven histogram equalization and community-adaptive link prediction, are summarized as follows.

4.1 Gray-Level Context-Driven Histogram Equalization

In this chapter, we proposed a new histogram equalization method to overcome the limitation of the previous gray-level partition based histogram equalizations, which were not able to equalize a sub-histogram crossing a partition boundary. Precisely we devised two enabling techniques called *mapping range* and *mapping distance* to make a full exploitation of mapping flexibility of gray-levels. With the integration of the two enabling techniques, we formulated the histogram equalization problem into a flow optimization problem in a network and solved it globally and efficiently. In addition, it was shown that the factor of minimizing power consumption was seamlessly combined to our network flow optimization framework. In conclusion, through experiments with the diverse images, it is shown that our proposed gray-level i.e., fine-grained context-driven equalization solution is able to achieve much more improved image quality, even with power saving, under the brightness preservation of images in comparison

with the images produced by the existing equalization methods. The implementation of histogram equalization without over-enhancement through the network flow method used in the proposed algorithm is also an important contribution to the image processing area.

4.2 Community-Adaptive Link Prediction in Social Networks

In this work, we proposed a new community-adaptable link prediction method to overcome the limitation of the previous community-aware methods, in which the similarity scoring was not tightly taken into account to the prior information of the community influence on link generation. To reflect the degree of the community influence, we separate link similarity scorings, depending on whether the links are inter or intra-links under community scope. Then, by utilizing the prior data on the link evolvement in the network, we were able to find a breakeven balancing weight between the scores of inter and inter-links to predict the top future links from all the missing links. Through experiments with the diverse datasets, it was shown that our proposed community-adaptive link prediction strategy could improve the prediction accuracy over the conventional community-aware link prediction algorithms that resorted to the same equation for similarity scores for both intra and inter-links.

Bibliography

[1] M. Chen, S. Mao, and Y. Liu, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, Jan. 2014.

[2] C. P. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Information Sciences*, vol. 275, pp. 314–347, Aug. 2014.

[3] C. Guillemot and P. Le Meur, *Digital image processing,* Pearson Education India, 2009.

[4] B. Furht, *Handbook of social network technologies and applications*, Springer Science & Business Media, 2010.

[5] L. Getoor and C. P. Diehl, "Link mining: a survey", *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, Dec. 2005.

[6] L. Lü and T. Zhou, "Link prediction in complex networks: A survey", *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, Oct. 2010.

[7] P. R. S. Soares and R. B. C. PrudeNcio, "Proximity measures for link prediction based on temporal events", *Expert Systems with Applications*, vol. 40, no. 16, pp. 6652–6660, Nov. 2013.

[8] D. Liben-Nowell and J. Kleinberg, "The Link-Prediction Problem for Social Networks," *Journal of the American society for information science and technology,* vol. 58, no. 7, pp. 1019–1031, March. 2007.

[9] Y. T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 1, pp. 1–8, Oct. 1997.

[10] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 68–75, Feb. 1999.

[11] S. D. Chen, and A. R. Ramli, "Minimum mean brightness error bi-histogram equalization in contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 40, no.4, pp. 1310–1319, Nov. 2003.

[12] M. A. A. Wadud, M. H. Kabir, M. A. A. Dewan, and O. Chae, "A dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, Feb. 2007.

[13] H. Ibrahim and N. S. P. Kong, "Brightness preserving dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1752–1758, Nov. 2007.

[14] D. Sheet, H. Graud, A. Suveer, M. Mahadevappa, and J. Chatterjce, "Brightness preserving dynamic fuzzy histogram equalization," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 2475–2480, Nov. 2010.

[15] T. Celik and T. Tjahjadi, "Automatic image equalization and contrast enhancement using gaussian mixture modeling," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 145–156, Jan. 2012.

[16] K. Singh and R. Kapoor, "Image enhancement using exposure based sub image histogram equalization," *Pattern Recognition Letters*, vol. 36, pp. 10–14, Jan. 2014.

[17] K. Singh and R. Kapoor, "Image enhancement via median-mean based sub-image-clipped histogram equalization," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 17, pp. 4646–4651, Sep. 2014.

[18] K. Singh, R. Kapoor, and S. K. Sinha, "Enhancement of low exposure images via recursive histogram equalization algorithms," *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 20, pp. 2619–2625, Oct. 2015.

[19] K. Gu, G. Zhai, S. Wang, M. Liu, J. Zhou, and W. Lin, "A general histogram modification framework for efficient contrast enhancement," *IEEE International Symposium on Circuits and Systems*, pp. 2816–2819, 2015.

[20] K. Gu, G. Zhai, X. Yang, W.Zhang, and C. W. Chen, "Automatic contrast enhancement technology with saliency preservation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1480–1494, Sep. 2015.

[21] K. Gu, G. Zhai, W. Lin, and M. Liu, "The analysis of image contrast: from quality assessment to automatic enhancement" *IEEE Transactions on Cybernetics*, 2015 (to appear).

[22] T. Celik and T. Tjahjadi, "Contextual and variational contrast enhancement," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3431–3441, Dec. 2011.

[23] Y. R. Lai, K. L. Chung, C. H. Chen, G. Y. Lin, and C. H. Wang, "Novel mean-shift based histogram equalization using textured regions," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2750–2758, Feb. 2012.

[24] S. Poddar, S. Tewary, D. Sharma, V. Karar, A. Ghosh, and S. K. Pal, "Non-parametric modified histogram equalization for contrast enhancement," *IET image Processing*, vol. 7, no. 7, pp. 641–652, Oct. 2013.

[25] C. Lee, C. Lee, Y. Y. Lee, and C. S. Kim, "Power-constrained contrast enhancement for emissive displays," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 80–93, Jan. 2012.

[26] C. Lee, C. Lee, and C. S. Kim, "Power-constrained contrast enhancement for OLED displays," *IEEE International Conference on Image Processing*, pp. 1689–1692, 2010

[27] O. Ghita; D.E. Ilea, and P.F. Whelan, "Texture enhanced histogram equalization using TV-L^1 image decomposition," *IET Image Processing*, vol. 7, no. 7, pp. 641–652, Oct. 2013.

[28] J.-Y. Kim, L.-S. Kim, and S.-H Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 475–484, April 2001.

[29] S. M. Pizer, *et al.*, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, pp. 355–368, 1987.

[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithm,* 2nd Edition, MIT Press, 2001.

[31] M. Dong, Y. S. K. Choi, and L. Zhong, "Power modeling of graphical user interfaces on OLED displays," *IEEE/ACM Design Automation Conference*, pp. 652–657, 2009.

[32] Kodak, http://r0k.us/graphics/kodak/, accessed January 2015.

[33] USC-SIPI, http://sipi.usc.edu/database/misc.zip, accessed January 2015.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[35] K. Gu, G. Zhai, X. Yang, W. Zhang, and M. Liu, "Structrual similarity weighting for image quality assessment," *IEEE International Conference on Multimedia and Expo Workshops*, pp. 1–6, 2013.

[36] S. S. Agaian, P. Karen, and A. M. Grigoryan, "Transform-based image enhancement algorithms with performance measure," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 367–382, March 2001.

[37] M. E. Newman, "Clustering and preferential attachment in growing network," *Physical review E*, vol. 64, no. 2, 025102, Jul. 2001.

[38] T. Zhou, L. Lü, and Y. C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, Oct. 2009.

[39] S. Soundarajan, and J. Hopcroft, "Using community information to improve the precision of link prediction methods," *In Proceedings of the 21st International Conference on World Wide Web*, pp. 607–608, Apr. 2012.

[40] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[41] J. Valverde-Rebaza and A. de Andrade Lopes, "Exploiting behaviors of communities of twitter users for link prediction," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1063–1074, Oct. 2013.

[42] J. Valverde-Rebaza and A. de Andrade Lopes, “Link prediction in complex networks based on cluster information,” *In Advances in Artificial Intelligence-SBIA 2012* , Springer Berlin Heidelberg, pp. 92–101, 2012.

[43] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical review E*, vol. 76, no. 3, 036106, Sep. 2007.

[44] J. Ding, L. Jiao, J. Wu, Y. Hou, and Y. Qi, “Prediction of missing links based on multi-resolution community division” *Physica A: Statistical Mechanics and its Applications*, vol. 417, pp. 76–85, Jan. 2015.

[45] Z. Li, S. Zhang, R. Wang, X. S. Zhang, and L. Chen, “Quantitative function for community detection,” *Physical review E*, vol. 77, no. 3, 036109, Mar. 2008.

[46] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, P10008, Oct. 2008.

[47] C. Lee, C. Lee, and C. S. Kim, “Power-constrained contrast enhancement for OLED displays,” *IEEE International Conference on Image Processing*, pp. 1689–1692, 2010

[48] M. Ley, “The DBLP computer science bibliography: Evolution, research issues, perspectives,” *International Symposium on String Processing and Information Retrieval*, Springer Berlin Heidelberg., pp. 1–10, Sep. 2002.

[49] J. J. Mcauley and J. Leskovec, “Learning to Discover Social Circles in Ego Networks,” *NIPS ,*, pp 548–56, Dec. 2012.

[50] J. Leskovec, J. Kleinberg, and C. Faloutsos “Graph evolution: Densification and shrinking diameters,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, Mar. 2007.

[51] KONECT, http://konect.uni-koblenz.de/networks/petster-friendships-hamster, accessed September 2016.

[52] J. Leskovec, D. Huttenlocher, and J. Kleinberg, “Signed networks in social media,” *Proceedings of the SIGCHI conference on human factors in computing systems* , pp. 1361–1370, Apr. 2010.

국문 초록

데이터 프로세싱 기법은 데이터를 분석 및 변환하여 우리가 원하는 결과를 얻는 방법이다. 본 연구에서는 이미지 프로세싱과 소셜 네트워크 분석 분야에 집중하여 데이터 프로세싱 기법을 적용한다. 이미지 분야에서는 그레이 레벨 맥락적 히스토그램 평활화 기법을 제안하며, 소셜 네트워크 분석 분야에서는 커뮤니티 적응형 링크 예측 기법을 제안하였다. 각각의 두 가지 분야에 대한 데이터 프로세싱 기법 연구에 관한 소개는 다음과 같다.

첫째, 히스토그램 평활화는 이미지의 그레이 레벨 분포를 재조정하는 기법으로써 이미지 화질을 개선하는 이미지 프로세싱에서 중요한 과정이다. 현재까지 굉장히 많은 히스토그램 평활화가 제안되었으며, 그 중 대다수 알고리즘들은 히스토그램을 몇 개의 섭히스토그램으로 나눈 후에 각각을 프로세싱하여 나중에 결과를 취합하는 방식을 사용하고 있다. 그러므로 해당 연구들은 어떤 방식으로 섭히스토그램을 나누는가에 초점을 맞췄다. 하지만, 파티션 기반 알고리즘들은 섭히스토그램 간의 간섭을 허용하지 않기 때문에 이미지 왜곡 등의 근본적인 문제가 발생한다. 이를 해결하기 위해서, 필자는 그레이 레벨 맥락적 히스토그램 평활화 기법을 제안한다. 간략히 말하자면, 히스토그램에서 그레이 레벨의 범위를 나누지 않고 다음의 두가지 기법을 적용하였다. (1) 범위로 나누지 않는 각각의 그레이 레벨을 위한 맵핑 범위. (2) 인접한 그레이 레벨간의 맵핑 유연성을 최대한 이용하기 위한 맵핑 거리. 그리고 우리는 위의 두 가지 기법을 활용하여 히스토그램 평활화 문제를 풀 수 있는 특별한 네트워크 구조의 플로우 최적화 문제를 공식화한 후 이를 효과적으로 풀었다. 또한, 우리의 네트워크 플로우 최적화 공식에 전력 소비 계수를 추가함으로써 이미지 화질과 전력 절약 간의 간단한 트페이드-오프 관계를 조절하였다.

둘째, 링크 예측은 소셜 네트워크 분석에서 중요한 연구 분야 중 한 가지이다. 링크 예측은 현재 시점에서는 연결되어 있지 않지만 미래 시점에서 연결될 것이라고 판단 되는 노드 쌍들을 찾는 문제이다. 예측의 정확도를 높이기 위해서 많은 연구들이 소셜 네트워크 구조내의 커뮤니티 정보를 고려하고자 시도하였다. 커뮤

니티 인식 링크 예측 알고리즘 중 대표적인 한 가지 전략은 링크가 구조적으로 같은 커뮤니티에 포함 여부를 표현하는 간단한 항을 통합된 링크 예측 수식에 포함하는 것이다. 하지만 단일항을 간단히 수식에 포함하는 방식은 커뮤니티의 구조에만 의존하기 때문에 소셜 네트워크에 따라서 비슷한 구조의 커뮤니티라 할지라도 링크 예측에 다르게 영향력을 미칠 수 있다는 점을 간과하였다. 이런 한계를 극복하기 위해서 커뮤니티에 따른 인트라와 인터 링크 두 가지로 구분하여 링크 예측을 하고, 커뮤니티가 링크 예측에 미치는 영향에 따라서 두 링크에 균형을 유지하는 커뮤니티 적응형 링크 예측 기법을 제안하였다.

결론적으로, 다양한 데이터 셋의 실험을 토대로 우리가 제안하는 이미지 프로세싱 분야의 그레이 레벨 맥락적 히스토그램 평활화와 소셜 네트워크 분석 분야의 커뮤니티 적응형 링크 예측 알고리즘이 기존의 방법들과 비교하여 더 나은 성능을 발휘하는 것을 보일 수 있었다.

주요어: 데이터 프로세싱, 이미지 프로세싱, 히스토그램 평활화, 소셜 네트워크 분석, 링크 예측

학번: 2013-30258